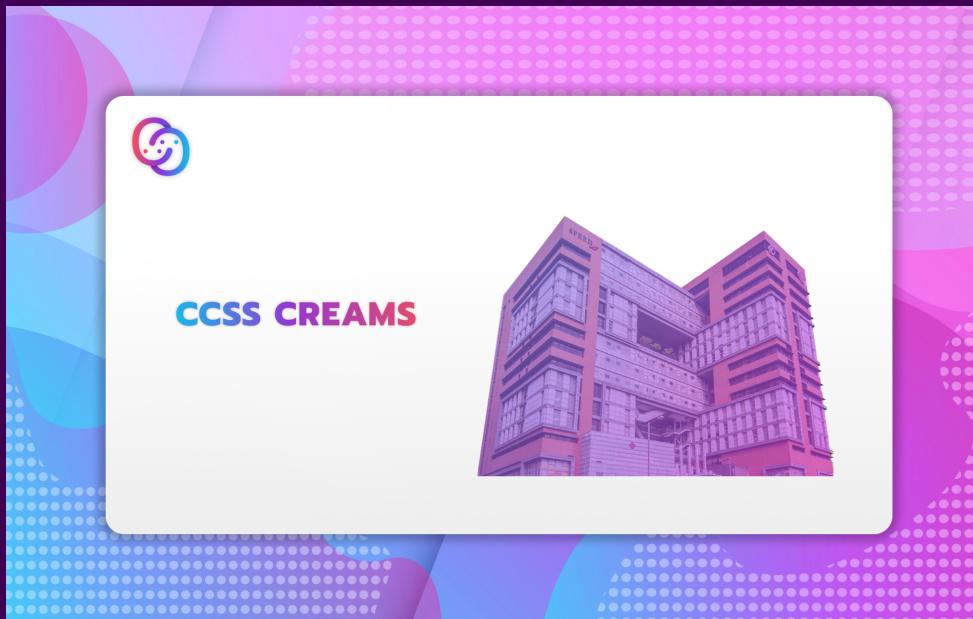




CCSS CREAMS

CCSS CREAMS Report



SEHH2240 201 Group B3

KOK Siu Chung Jesse 20190875A***

HO Chun Hei Stanley 20096776A

MOK Ho Ming Ricky 20072190A

TO Fui Yin Richard 20199034A

16-4-2021

1. A Detailed Description of The Idea.	3
2. Describe business rules	4
Who will use the system?	4
3. Conceptual Model (ERD)	7
4. Logic design (Form)	17
(a) The login page	22
(b) Student side	26
(c) Teacher side	95
6. Logic design (Reports)	124
7. Logic design (Queries)	137
(a) Normal Search Queries	140
(b) Linked Search Queries	152
(c) Linked Insert Queries	164
(d) Normal Insert Queries	166
8. Implementation Details	168
9. Difficulties	169
(a) Design phase	169
(b) Implementation Phase	172
(c) Lesson learnt	174
10. Work distribution list	175
11. Conclusion	176
12. Reference	177

1. A Detailed Description of The Idea.

CC Secondary School is a secondary school in Hong Kong. The school focused on developing students as a “whole-person”, they provided many extracurricular activities for students. In order to keep track of the participants of all the activities. However, students are always unaware of the announcement of the extracurricular activities. Sometimes, when teachers ask them why they don’t participate in those activities, a lot of the time they would say that they didn’t even know there was a particular activity in the first place.

That’s why the school hired us to design a database system for them, just to **inform students** what they can **participate in, register those activities** in the database system. So that they can enjoy their secondary school life. The system’s primary goal is to **store activities and club data** for them.

The name of our database is **CREAMS**, which stands for our system is called CREAMS, which stands for

Co-curricular and Club

Registration

Enrollment

And

Management

System

2. Describe business rules

Who will use the system?

- Students
- Teachers

Students: Only read, add and modify their own activity

Teachers: Can read, add delete and modify student's record

Students can enter their choice of activity.

If the student is a chairman of a club he can represent the club to add activities.

We would like to generate a SLP for students and that's why we consider adding store the academic results and self accounts in the database system.

First, we have a table **student_information** which stores student information.

Then, as we need to make a login system for students, we make a student login system and each student has their own student login password so the relationship between **student_information** and **student_login** is **1:1 relationship**.

Some students may like to write a self account of their own and each student will only have one self account, that's why the relationship between **student_information** and **student_selfAccount** is **1:1 relationship**.

We need to store student grades in each year as this system can be used cross-year and student's grade, class and class number may change. That's why the relationship between **student_information** and **student_grade** is **1:M relationship**.

A student will have different academic results in different school years and they will have different academic results in different semesters. That's why

the relationship between **student_information** and **student_academic** is a **1:M relationship**.

A club may run for several years so that the relationship between **club_information** and **club_schoolYear** is **1:M relationship**.

A student can enter different clubs in a particular year so that the relationship between **student_inforamtion** and **club_schoolYear** is **M:M relationship**. That's why we separate it into two different 1:M relationships. We create a table called Club_record, the relationship between **student_inforamtion** and **club_record** is **1:M relationship** and the relationship between **club_schoolYear** and **club_record** is the other **1:M relationship**.

A student can be a chairman of a certain club so that the relationship between **student_information** and **club_schoolYear** is **1:M relationship**.

As student may have different medical situations, that's why the relationship between **student_information** and **student_medical** is **1:M relationship**

As almost all of the secondary students are under 18, something that they cannot decide by themselves, and this is why there is a **student_guardian** table. As there are different versions of guardian information, that means that the relationship between **student_information** and **student_guardian** is **1:M relationship**.

A student can have many awards or no awards during his secondary school life. That's why the relationship between **student_information** and **student_awards** is **1:M relationship**.

As students can participate in different activities and also students can not participate in any activities, that's why the relationship between **student_inforamtion** and **activity_records** is **1:M relationship**.

Different activities have different participants and that will contain a lot of activity records so that the relationship between **activity_information** and **activity_records** is **1:M relationship**.

A club may hold many activities and an activity may be held by a club, this leads to the relationship between **club_record** and **activity_information** is **1:M relationship**.

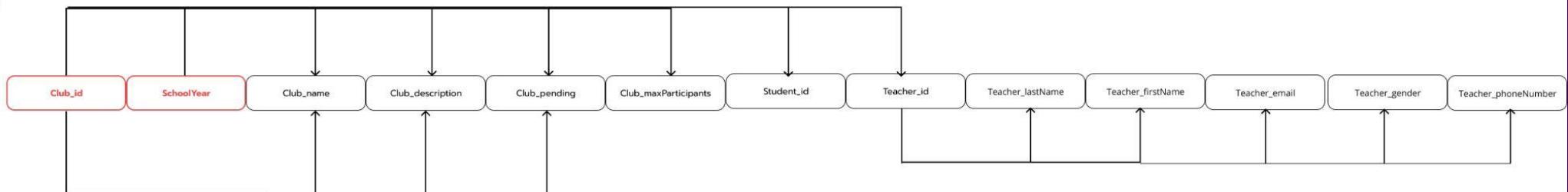
A teacher may be in charge of a couple clubs in different school years and a club must have one teacher in charge, this leads to the relationship between **teacher_information** and **club_schoolYear** is **1:M relationship**.

A teacher may hold many activities and each activity needs a teacher to be off, that's why the relationship between **teacher_information** and **activity_information** is **1:M relationship**.

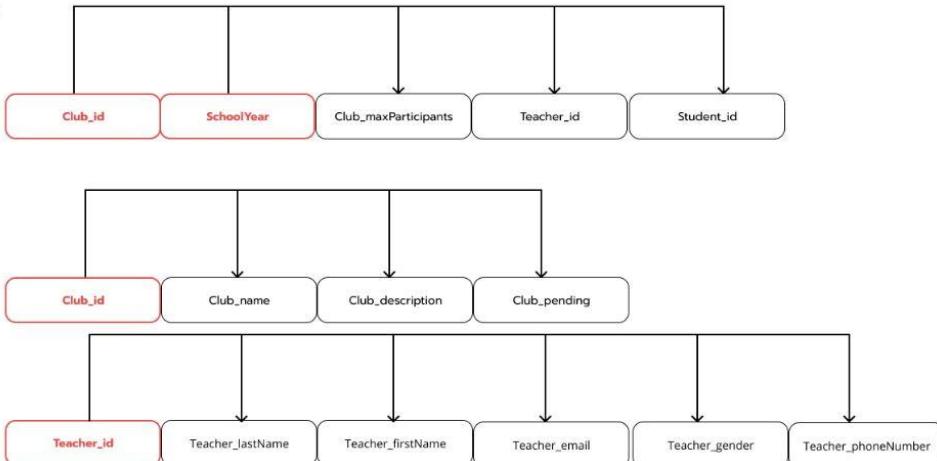
Each teacher has their own unique teacher id and with the login system they will have their own password. This is why the relationship between **teacher_information** and **teacher_login** is **1:1 relationship**.

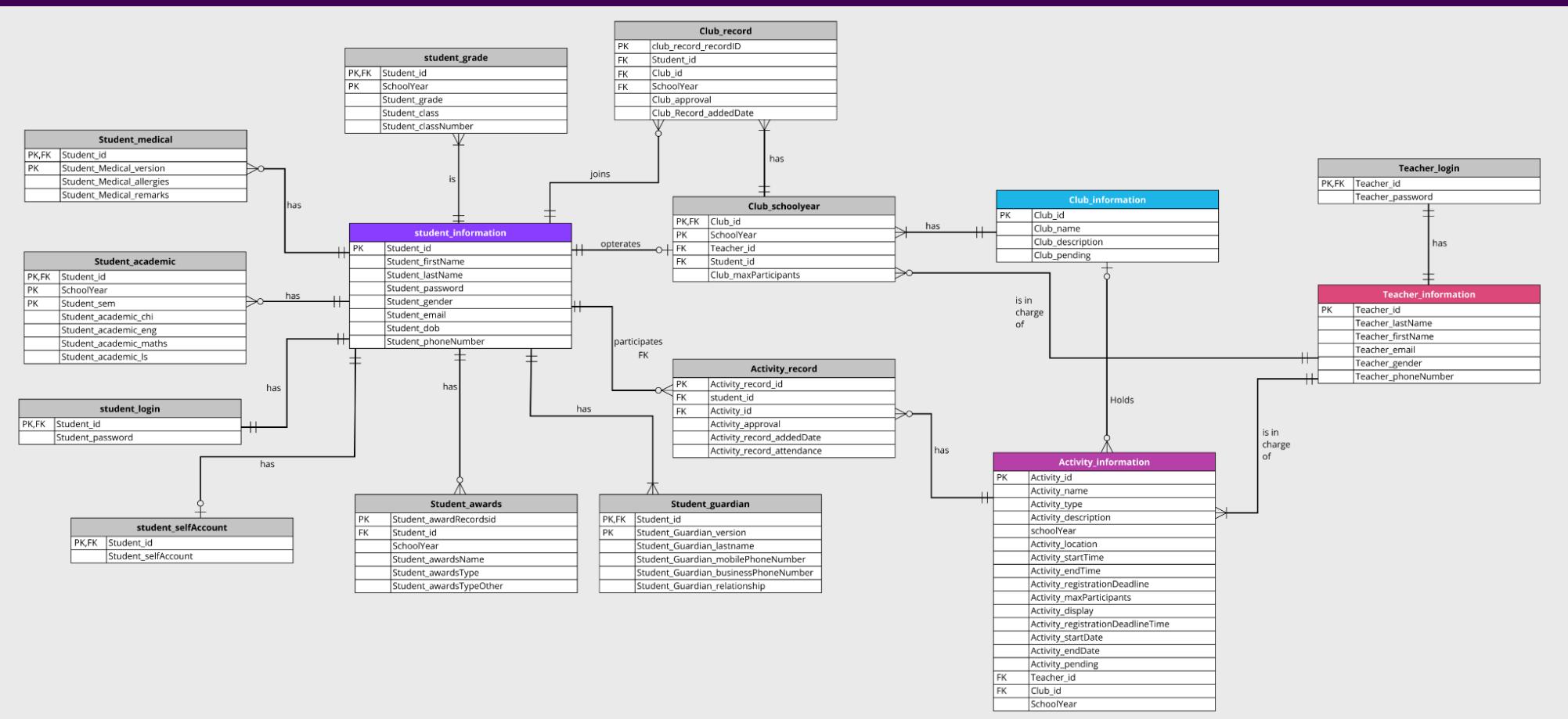
3. Conceptual Model (ERD)

1NF

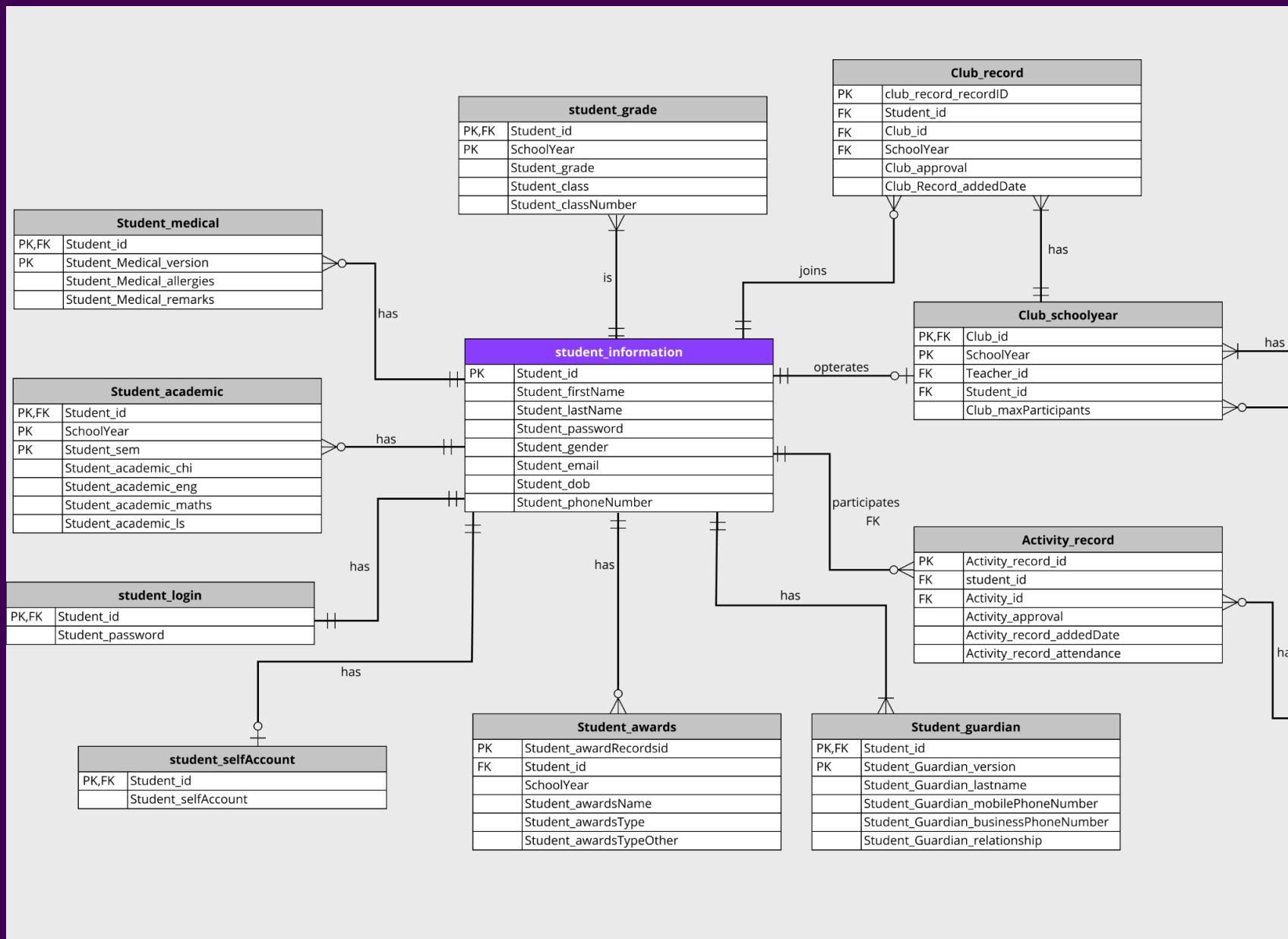


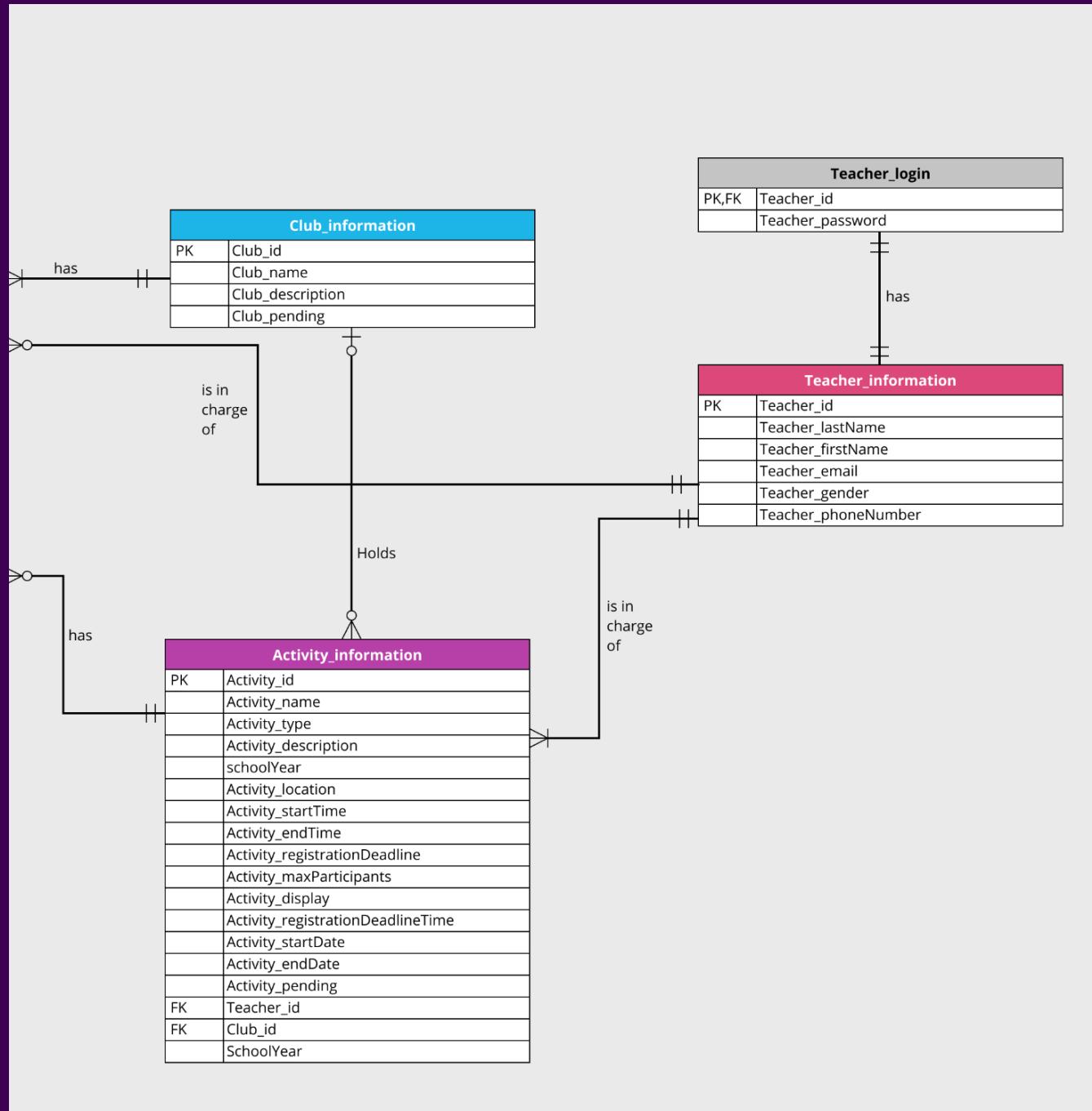
3NF





Close Up





Data dictionary and schema

Activity_information

Field	Data type	Description
Activity_id	Short Text	Uniquely identifies each activities
Club_id	Short Text	Link to club_id
Activity_type	Short Text	To tell what type of the activity e.g. workshops, competition, meeting
Activity_name	Short Text	Name of the activity
Activity_description	Long Text	Said what this activity is about
Activity_startDate	Date/Time	The date that the activity started
Activity_endDate	Date/Time	The date that the activity ended
Activity_startTime	Date/Time	The time that the activity started
Activity_endTime	Date/Time	The time that the activity ended
Activity_registrationDeadline	Date/Time	The registration of the activity deadline date
Activity_registrationDeadlineTime	Date/Time	The registration of the activity deadline time
Activity_maxParticipants	Number	Store the maximum number of participants
Teacher_id	Short Text	Linked to Teacher_id
Activity_display	Yes/No	Will the activity display to people
Activity_pending	Yes/No	Activity has to approve student to join
Activity_location	Short Text	Where the activity is held
SchoolYear	Date/Time	The year that this activity at

Activity_record

Field	Data type	Description
Activity_record_id	AutoNumber	Uniquely identifies each Activity record
student_id	Short Text	Linked to student_id
Activity_id	Short Text	Linked to Activity_id
Activity_approval	Yes/No	If yes, student is IN
Activity_record_addedDate	Date/Time	When this record had been added
Activity_record_attendance	Yes/No	To check if the student attend the activity

Club_information

Field	Data type	Description
Club_id	Short Text	the club id which can uniquely identify each club
Club_name	Short Text	The name of the club
Club_descripition	Long Text	Short introduction of each club
Club_pending	Yes/No	Activity has to approve student to join

Club_record

Field	Data type	Description
club_record_recordID	AutoNumber	Each the_record_recordID uniquely identifies each record
Student_id	Short Text	Linked to Student_id
Club_id	Short Text	Linked to Club_id
SchoolYear	Short Text	Which SchoolYear this club record is at
Club_approval	Yes/No	If YES, student is IN
Club_Record_addedDate	Date/Time	The date that this club record is added

Club_schoolyear

Field	Data type	Description
Club_id	Short Text	Linked to Club_id
SchoolYear	Short Text	Linked to SchoolYear
Teacher_id	Short Text	Linked to Teacher_id
Student_id	Short Text	Linked to Student_id
Club_maxParticipants	Number	The Max Participants of that club

Student_academic

Field	Data type	Description
Student_id	Short Text	Linked to Student_id
SchoolYear	Short Text	Which year in this record
Student_sem	Number	Linked Student_sem
Student_academic_chi	Number	Chinese academic result
Student_academic_eng	Number	English academic result
Student_academic_maths	Number	Mathematics academic result
Student_academic_ls	Number	Libraral Studies academic result

Student_awards

Field	Data type	Description
Student_award Recordsid**	AutoNumber	Uniquely identifies each record
Student_id	Short Text	Linked to Student_id
SchoolYear	Short Text	SchoolYear of this record at
Student_award sName	Long Text	The name of the award
Student_award sType	Short Text	The type of the award
Student_award sTypeOther	Short Text	Other additional information about the award

Student_grade

Field	Data type	Description
Student_id	Short Text	Linked to Student_id
SchoolYear	Short Text	Which school year this record in
Student_grade	Number	Which grade of the student
Student_class	Short Text	The class of student
Student_classN umber	Number	The class number of student

Student_guardian

Field	Data type	Description
Student_id	Short Text	Linked to Student_id
Student_Guardian_version	Number	The version of the guardian
Student_Guardian_lastname	Short Text	The name of student guardian
Student_Guardian_mobilePhoneNumber	Short Text	The mobile phone number of student guardian
Student_Guardian_businessPhoneNumber	Short Text	The business phone number of student guardian
Student_Guardian_relationship	Short Text	The relationship between student and the Guardian

Student_information

Field	Data type	Description
Student_id	Short Text	The student id which can uniquely identify each student
Student_firstname	Short Text	The First Name of the student
Student_lastname	Short Text	The Last Name of the student
Student_gender	Short Text	Is Male/Female?
Student_email	Short Text	Store student email so activity holder can have a way to contact participants
Student_dob	Date/Time	The date of birth of student
Student_phone_Number	Short Text	The Phone Number of the Student

Student_login

Field	Data type	Description
Student_id	Short Text	Linked to Student_id
Student_password	Short Text	The password of student

Student_medical

Field	Data type	Description
Student_id	Short Text	Linked to Student_id
Student_Medical_version	Number	The medical version of the record
Student_Medical_allergies	Short Text	Which allergies does the student have
Student_Medical_remarks	Long Text	Any additional information about this record

Student_selfAccount

Field	Data type	Description
Student_id	Short Text	Linked to Student_id
Student_selfAccount	Long Text	Student can type the self account

Teacher_information

Field	Data type	Description
Teacher_id	Short Text	The Teacher id which can uniquely identify each Teacher
Teacher_lastName	Short Text	The last name of teacher
Teacher_firstName	Short Text	The first name of teacher
Teacher_email	Short Text	The email of teacher
Teacher_gender	Short Text	The gender of teacher
Teacher_phoneNumber	Short Text	The phone number of teacher

Teacher_login

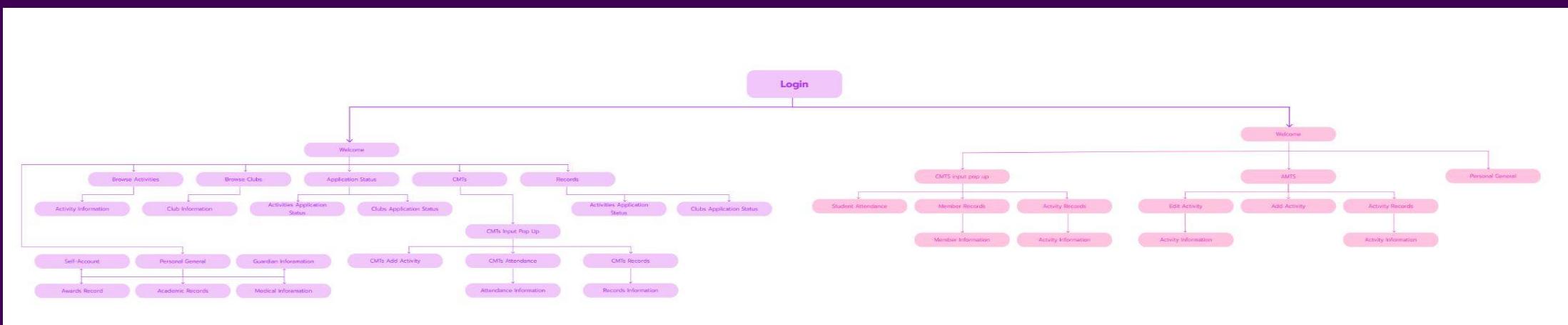
Field	Data type	Description
Teacher_id	Short Text	Linked to teacher id
Teacher_password	Short Text	The password of teacher

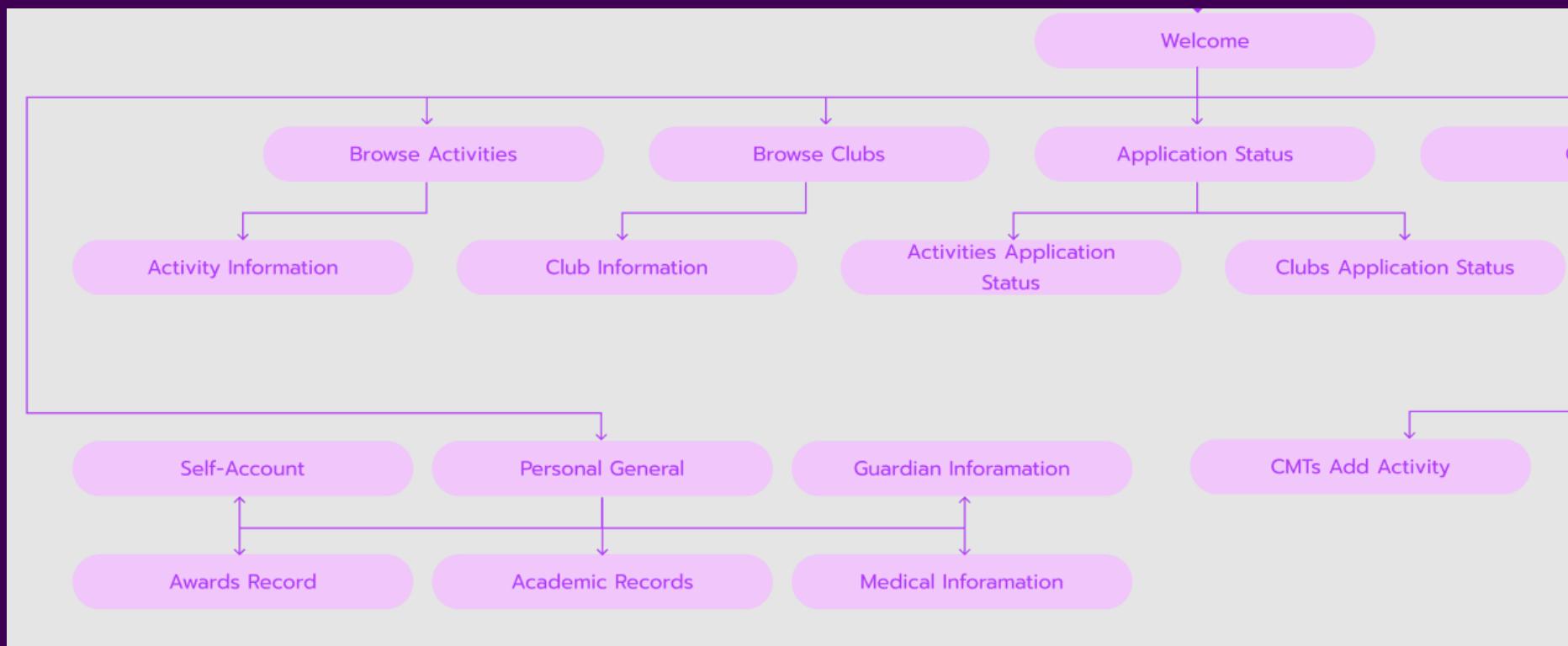
Teacher_login

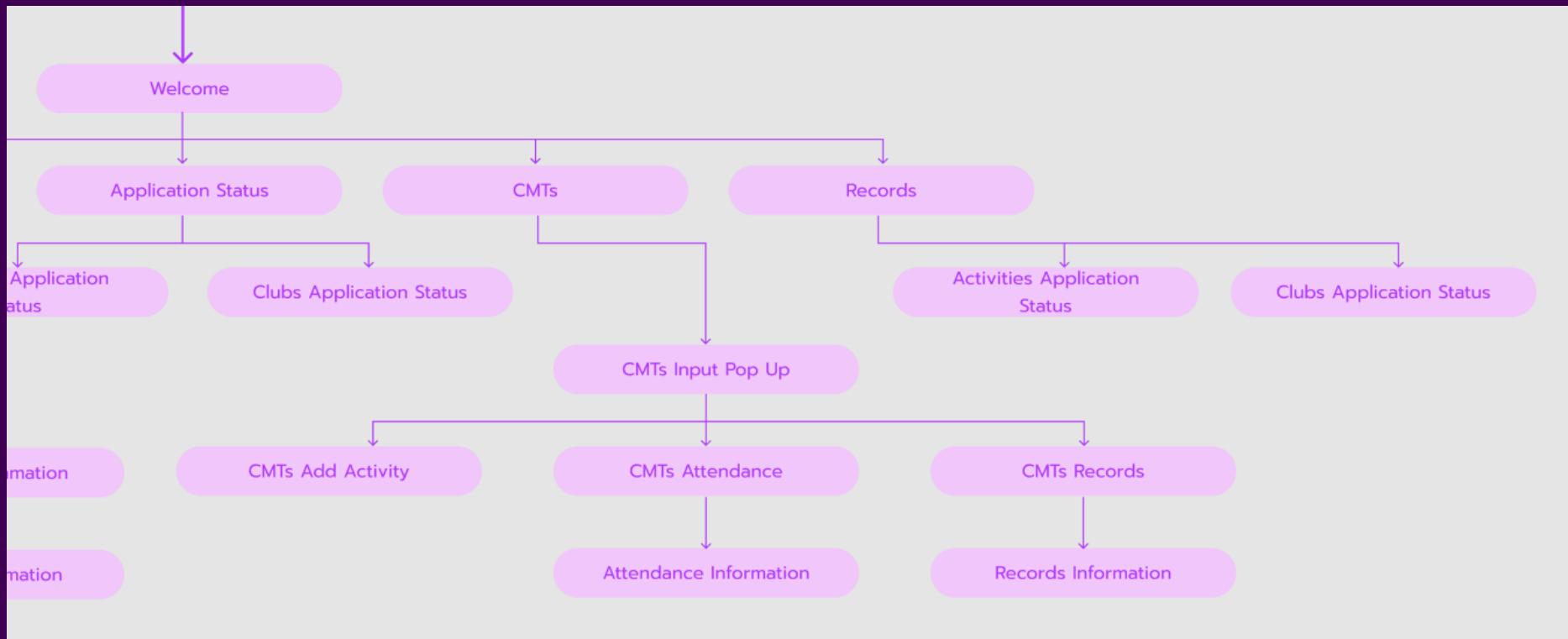
Field	Data type	Description
Teacher_id	Short Text	Linked to teacher id
Teacher_password	Short Text	The password of teacher

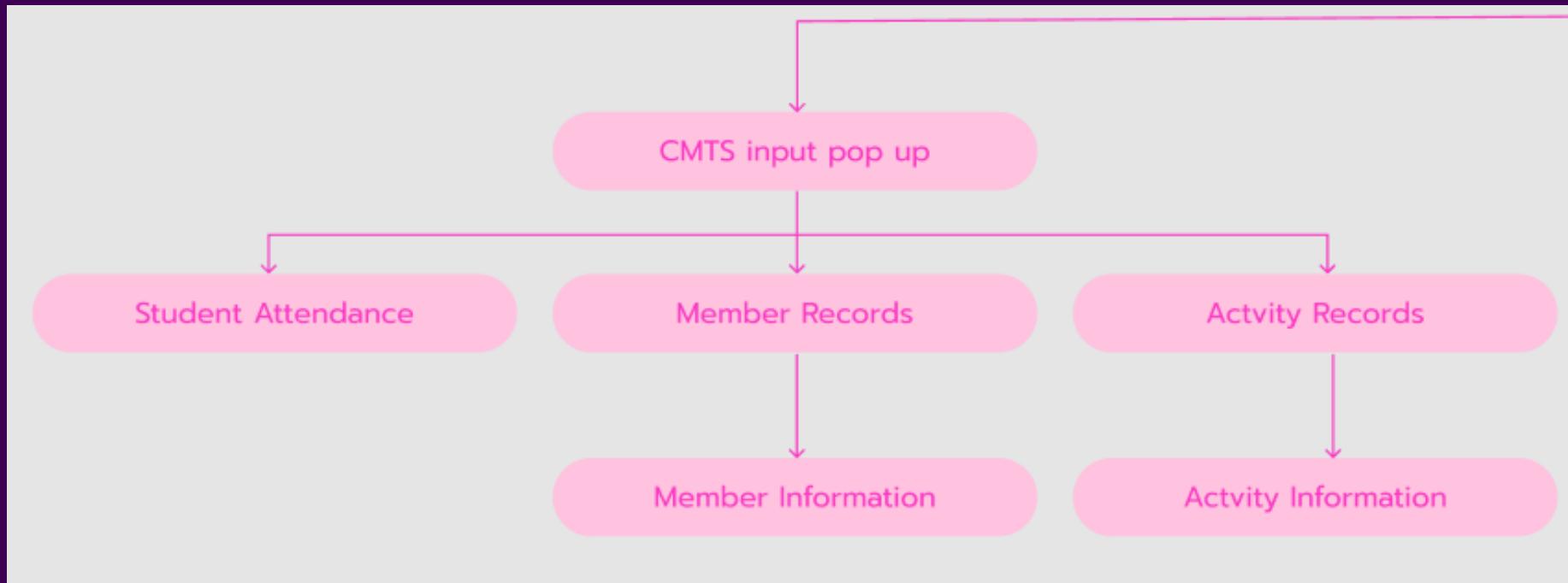
** field is the primary key. * is the foreign key

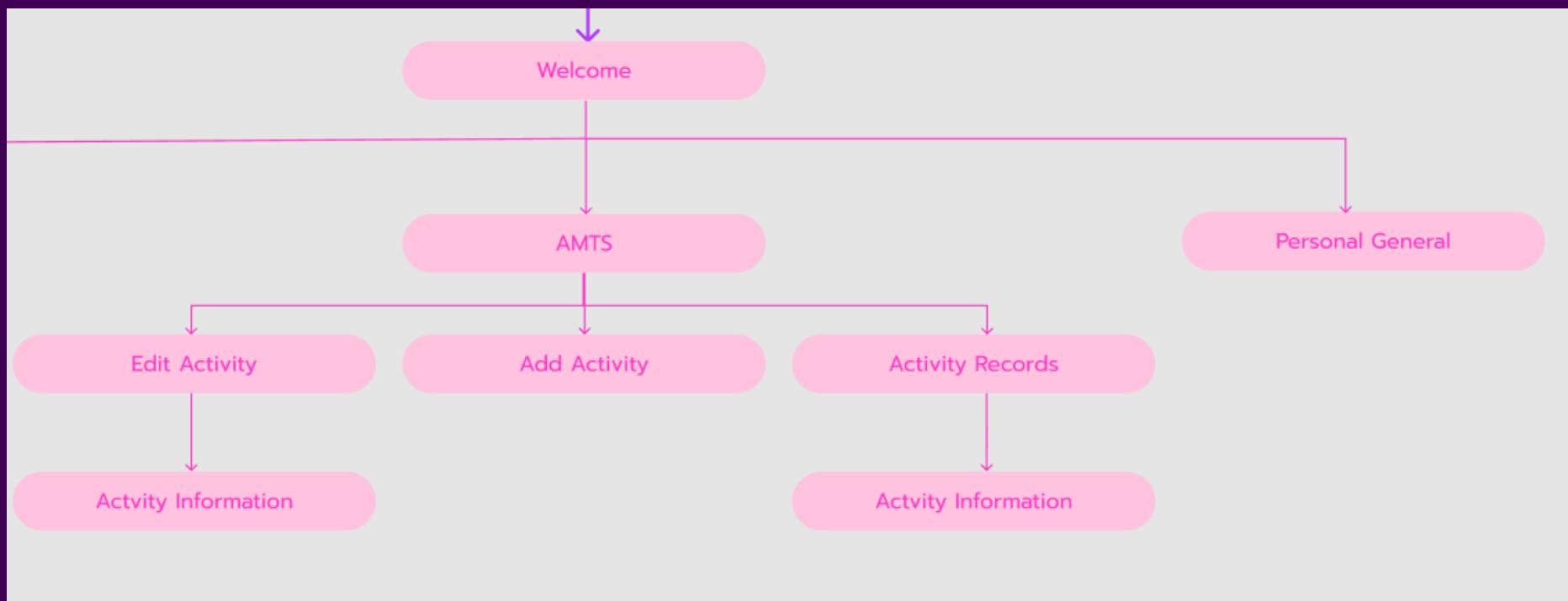
4. Logic design (Form)





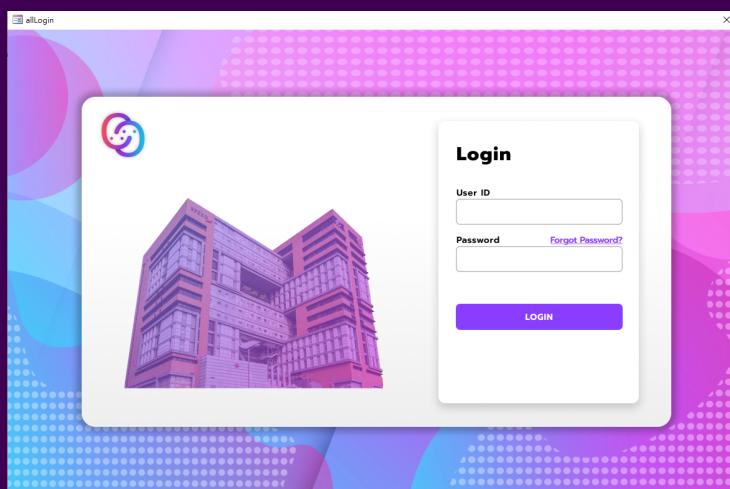




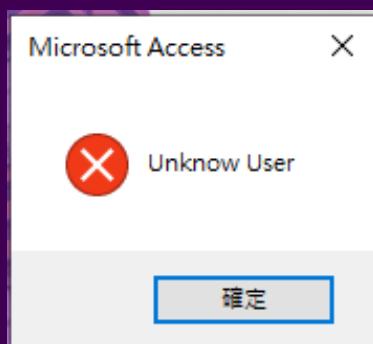
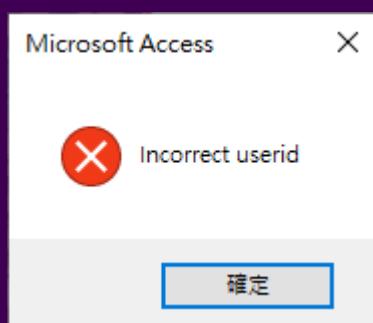


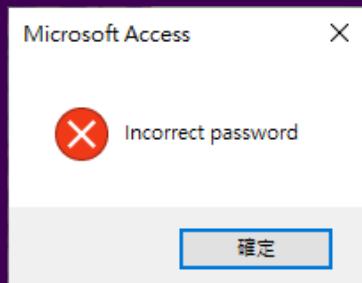
(a) The login page

As we want to make a legit system, for this we first need students and teachers to have their own id and password so that they can login to their own page using the system's service. To finish that we create a student login form and the form looks like this.



It is for all logins, teachers and students can all login using this form, why we can do that is because we use VBA to extract and check the first letter of the user ID if a **student** login it's **user ID** will start with '**S**' and **teacher login**, it's id will start as '**T**'. And when you input a user id that represents no one, an **error message will pop up**. Telling users that their inputs are incorrect.





It will react a certain way so that people will know what they type wrong.

VBA Code of "Form_allLogin"

```
Option Compare Database
Option Explicit

Public Currentid As String

Function PasswordCompare(Currentid_password As String) As Boolean

    If Currentid_password = Me.User_password Then
        PasswordCompare = True
    Else
        PasswordCompare = False
    End If

End Function

Private Sub btn_loginOverlay_Click()

    Dim Currentid_Code As String
    Dim Currentid_password As String

    If IsNull(Me.User_id) And IsNull(Me.User_password) Then
        MsgBox "Enter user id and password", vbInformation,
        "Empty field"
        Me.User_id.SetFocus
    ElseIf IsNull(Me.User_id) Then
        MsgBox "Enter a user id", vbInformation, "Empty field"
        Me.User_id.SetFocus
    ElseIf IsNull(Me.User_password) Then
        MsgBox "Enter a password", vbInformation, "Empty field"

    End If
End Sub
```

```
Me.User_password.SetFocus

Else
    setCurrentid (Nz(Me.User_id, 0))
    Currentid = getCurrentid
    Currentid_Code = Left(Currentid, 1)

'=====
=====

If (Currentid_Code = "T") Then
    Currentid_password = Nz(DLookup("Teacher_password",
"Teacher_login", "Teacher_id ='" & Me.User_id & "'", "0"))

    If Currentid_password = 0 Then
        ErrorMessage (1)

    Else
        If PasswordCompare(Currentid_password) Then
            ToPage "Teacher_welcome"

        Else
            ErrorMessage (2)

    End If

End If

'=====
=====

ElseIf (Currentid_Code = "A") Then
    Currentid_password = Nz(DLookup("Admin_password",
"Admin_information", "Admin_id ='" & Me.User_id & "'", "0"))

    If Currentid_password = 0 Then
        ErrorMessage (1)

    Else
        If PasswordCompare(Currentid_password) Then
            ToPage "Student_welcome"

        Else
            ErrorMessage (2)

    End If

End If
```

```
'=====
=====
ElseIf (Currentid_Code = "S") Then
    Currentid_password = Nz(DLookup("Student_password",
"Student_login", "Student_id ='" & Me.User_id & "'", "0"))

    If Currentid_password = 0 Then
        ErrorMessage (1)

    Else
        If PasswordCompare(Currentid_password) Then
            ToPage "Student_welcome"

        Else
            ErrorMessage (2)

        End If

    End If

Else
    ErrorMessage (3)

End If
End Sub

Private Sub Forgotpw_Click()
Dim str As String

    str =
"https://docs.google.com/forms/d/e/1FAIpQLSdx1N3pjLUBYBq9iUqhHAo2
9zC7H-xtWEqQ3YIcOSsvNpb7CA/viewform"

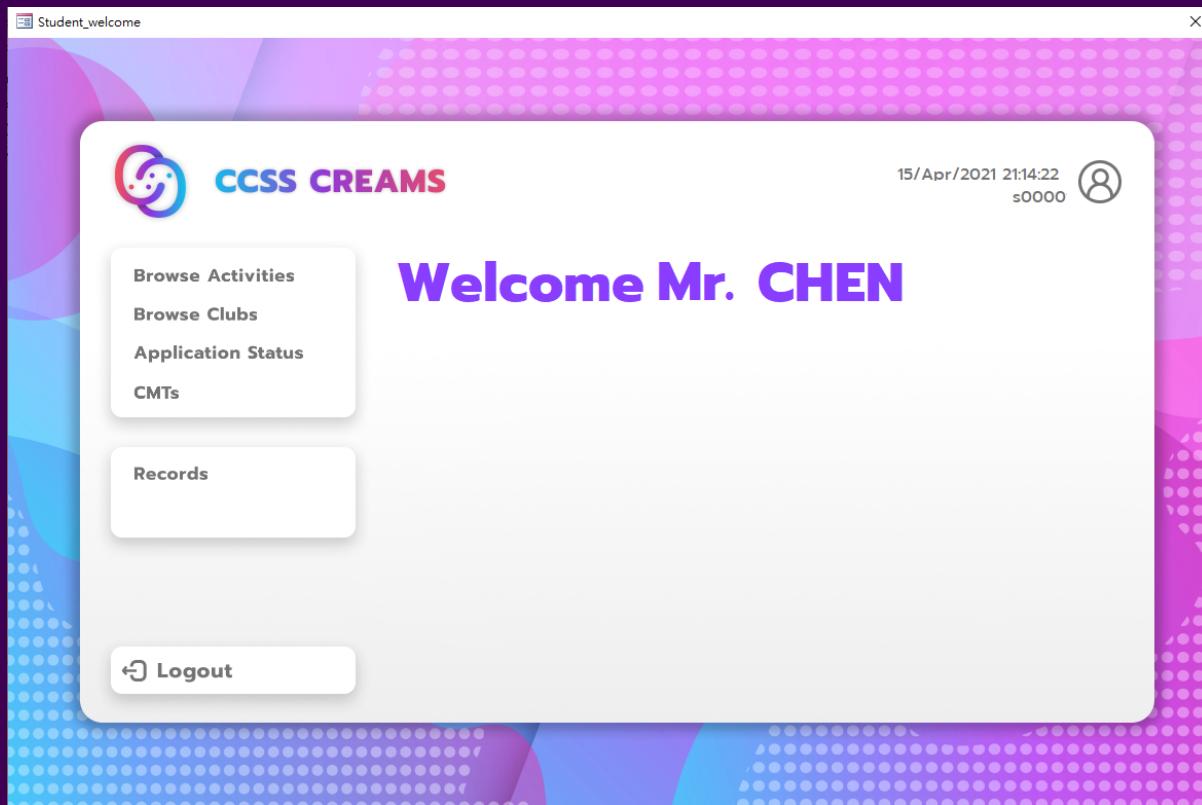
    Application.FollowHyperlink str

End Sub

Private Sub Form_Load()
    setCurrentid ("")
End Sub
```

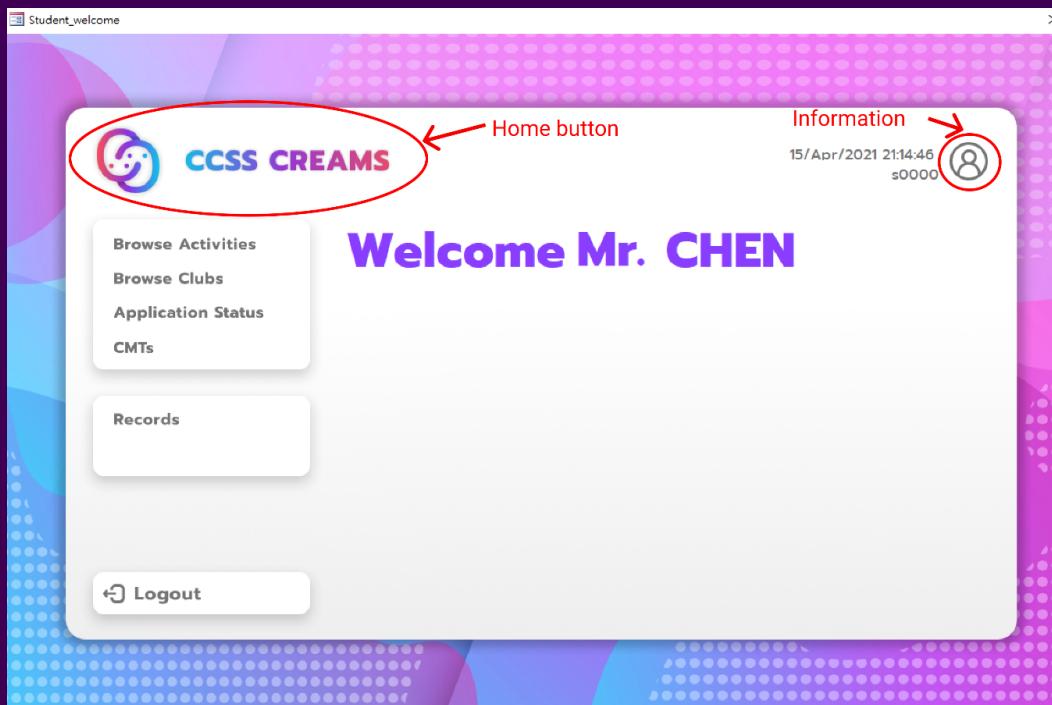
(b) Student side

After logging in with a student account, it will transfer us to "Student_welcome" Form, the main page of Student.

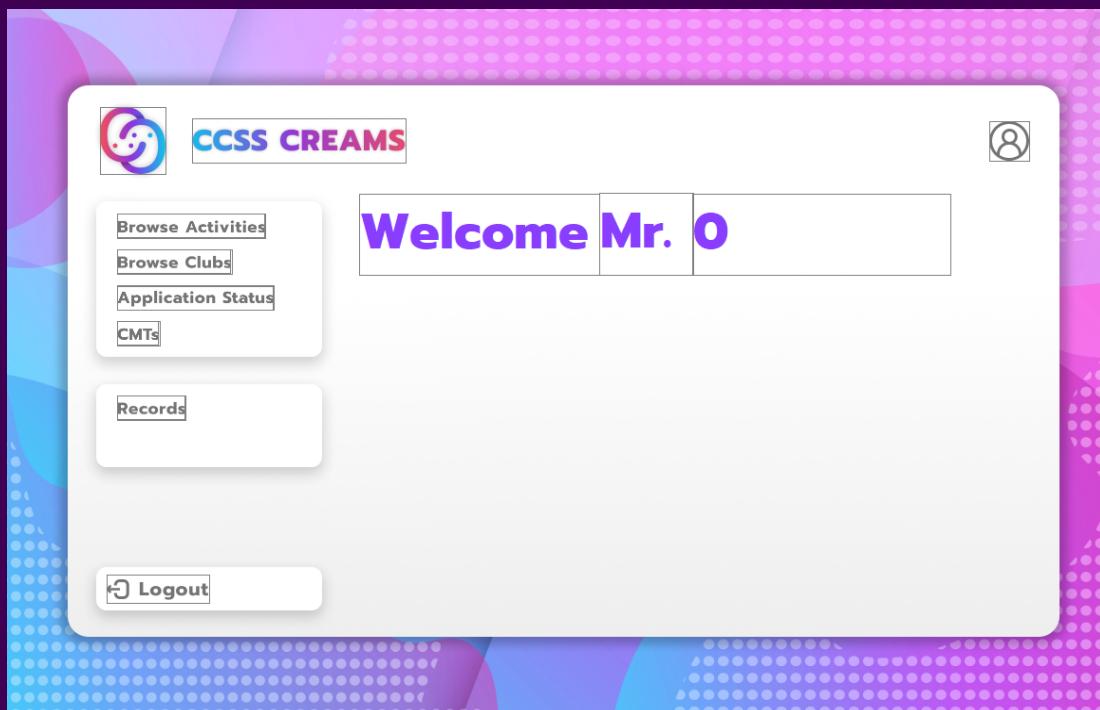


In this main page, there is a greeting message for students and different buttons for navigations and the greeting will get students or teacher's last name and display it in this case as s00001 is Chan Tai Man, It will display "Welcome Mr. Chan". Other than that, we have our system logo on the top left corner, and personal information logo on the top right corner. At the left hand side, there is a navigation menu, for students to navigate different forms. There are "Browse Activities", "Browse Clubs", "Application Status", "CMTs" and "Records".

Other than the navigation button on the left hand side, there are two buttons that people probably will miss out, And these buttons are "Personal" and "Home" that we will explain later.



Design View of this form:



VBA Code of this form

```
Option Compare Database

Private Sub btn_HomeLogoOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_HomeTextOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_LogoutOverlay_Click()

    ToPage "allLogin"
    setCurrentid ("")
End Sub

Private Sub btn_PersonOverlay_Click()

    ToPage "Student_Personal_general"

End Sub

Private Sub Form_Load()
    Dim CurrentStudentGender As String

    Me.txt_stdid.Caption = getCurrentid
    Me.std_lname.Caption = Nz(DLookup("[Student_lastName]",
    "Student_information", "[Student_id] = '" & [getCurrentid] &
    "'"), 0)
    CurrentStudentGender = Nz(DLookup("[Student_gender]",
    "Student_information", "[Student_id] = '" & [getCurrentid] &
    "'"), 0)

    If CurrentStudentGender = "F" Then
        Me.std_gender.Caption = "Ms."
    Else
        Me.std_gender.Caption = "Mr."
    End If

```

```
Me.txt_Clock.Caption = getNowDate & " " & getNowTime

End Sub

Private Sub Form_Timer()

    Me.txt_Clock.Caption = getNowDate & " " & getNowTime

End Sub

Private Sub btn_BrowseActivitiesOverlay_Click()

    ToPage "Student_Browseactivities"

End Sub

Private Sub btn_BrowseClubsOverlay_Click()

    ToPage "Student_browseClubs"

End Sub

Private Sub btn_ApplicationStatusOverlay_Click()

    ToPage "Student_applicationStatus"

End Sub

Private Sub btn_CMTsOverlay_Click()

    ToTempPage "Student_CMTs_popUP"

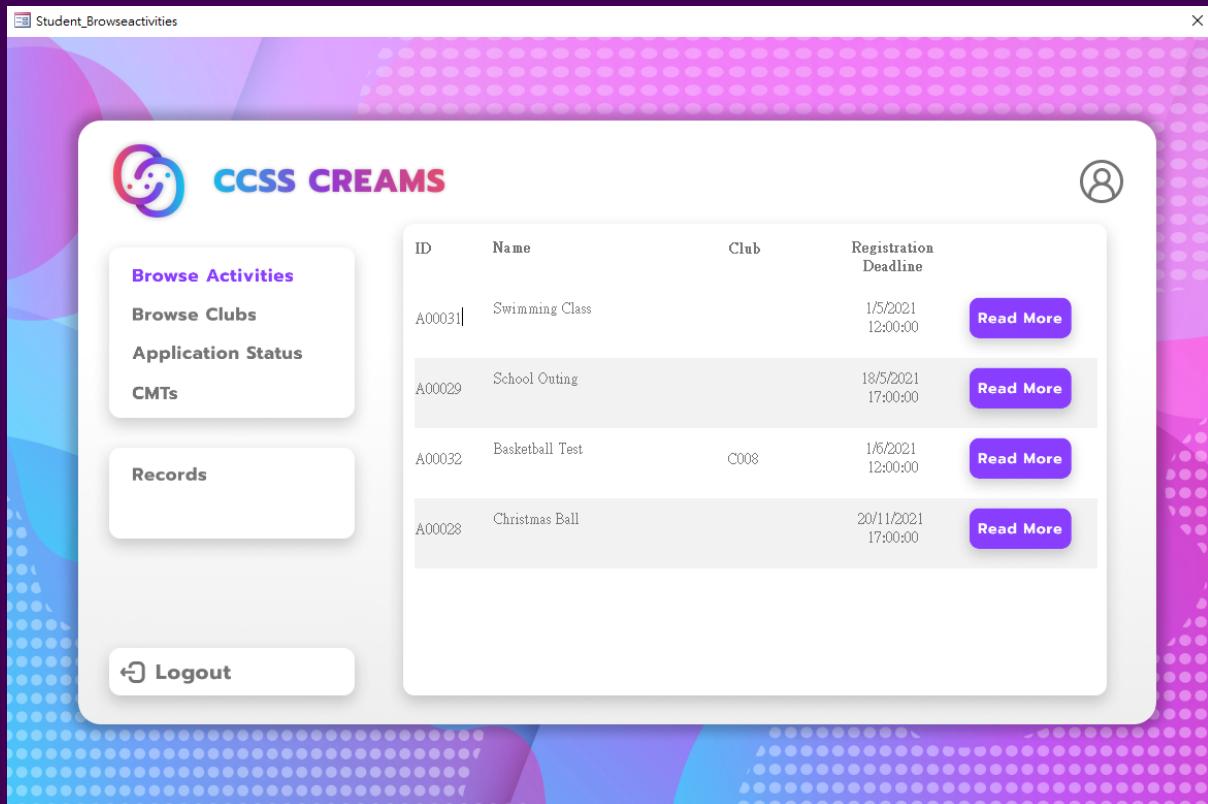
End Sub

Private Sub btn_recordsOverlay_Click()

    ToPage "Student_records"

End Sub
```

Students can click Browse activities in order to bring to another new form, which is student_Browseactivities form and this form will **show activities that are available to sign up** for. Which is the activity that the registration deadline is before now.



VBA CODE

```
Option Compare Database

Private Sub btn_HomeLogoOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_HomeTextOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_LogoutOverlay_Click()

    ToPage "allLogin"
    setCurrentid ("")
```

```
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

Private Sub Form_Load()
    Me.txt_stdid.Caption = getCurrentid
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.txt_schoolyear.Caption = getCurrentSchoolYear

    Me.Refresh
    DoCmd.Requery
End Sub

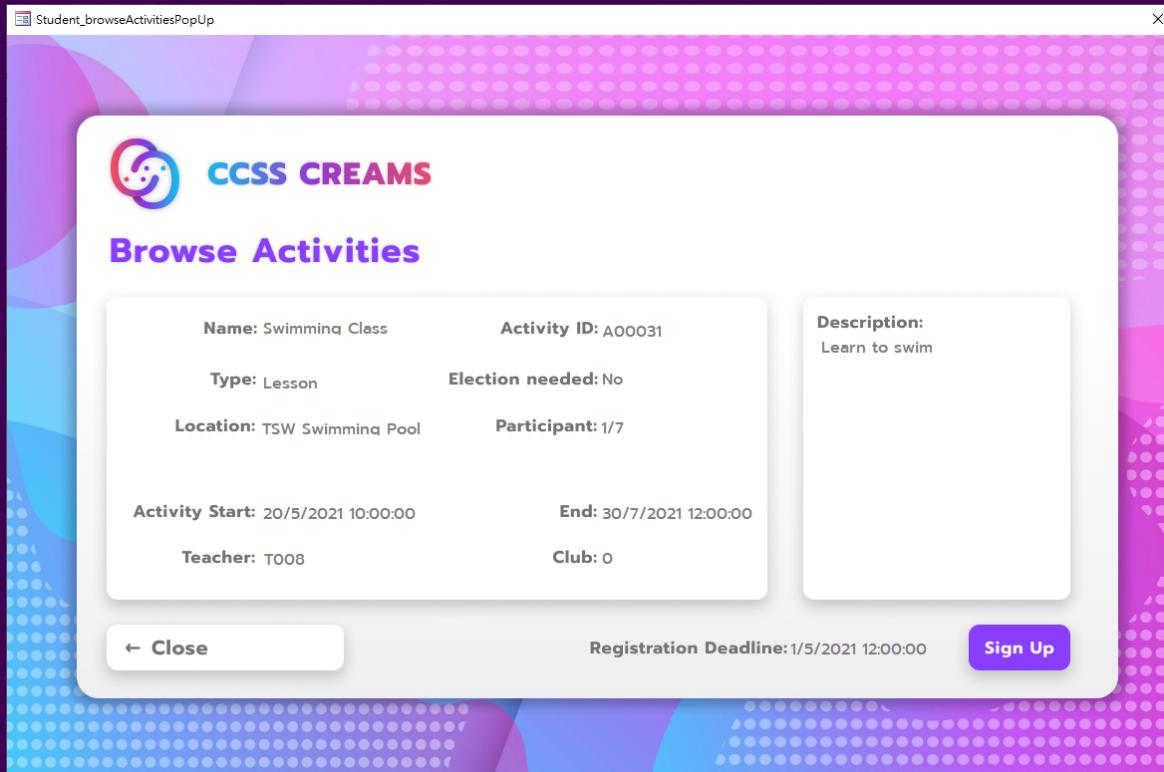
Private Sub btn_BrowseActiviesOverlay_Click()
    Me.Refresh
    DoCmd.Requery
End Sub

Private Sub btn_BrowseClubsOverlay_Click()
    ToTempPage "Student_browseClubs"
End Sub

Private Sub btn_ApplicationStatusOverlay_Click()
    ToTempPage "Student_applicationStatus"
End Sub

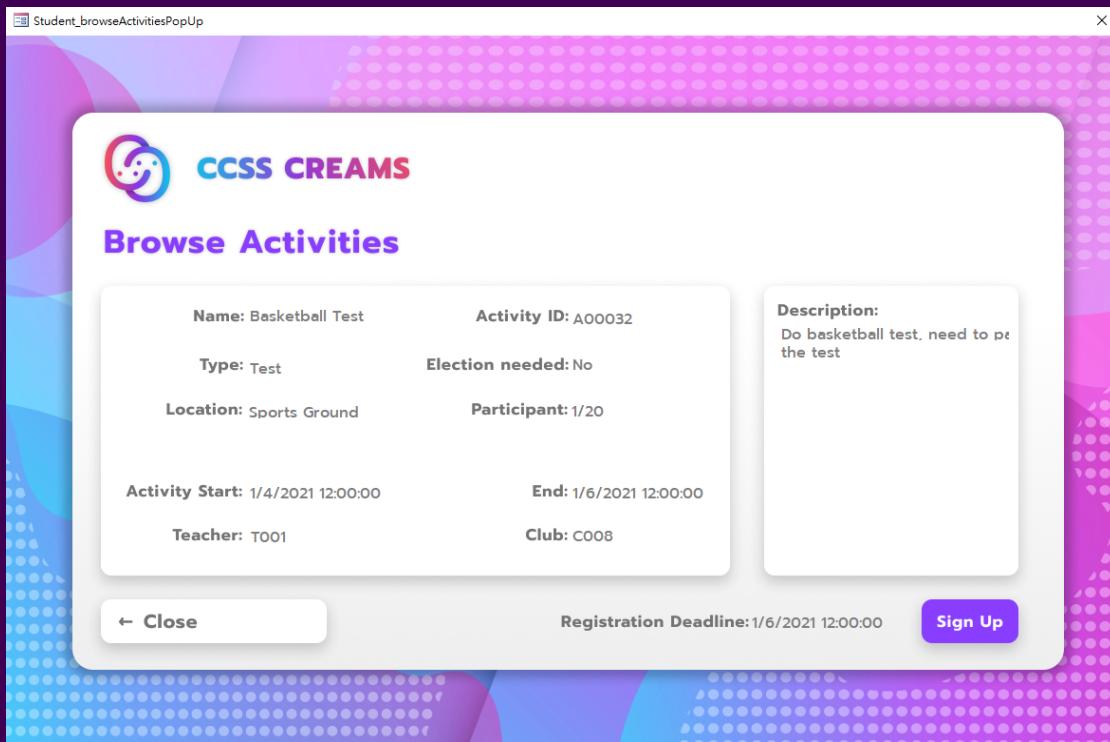
Private Sub btn_CMTsOverlay_Click()
    ToTempPage "Student_CMTs_popUP"
End Sub
```


When students are interested in one of the activities, they can click on read more to check more information about that particular activity.

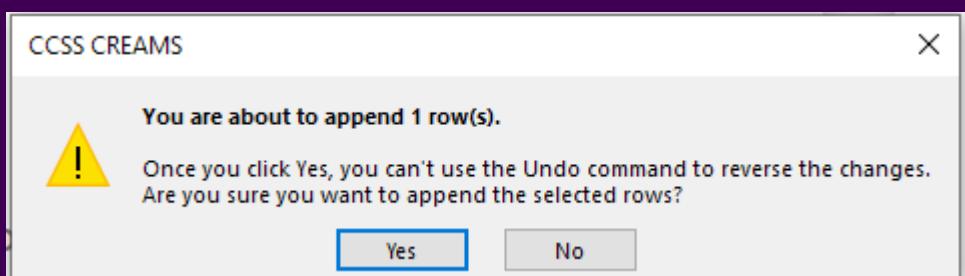
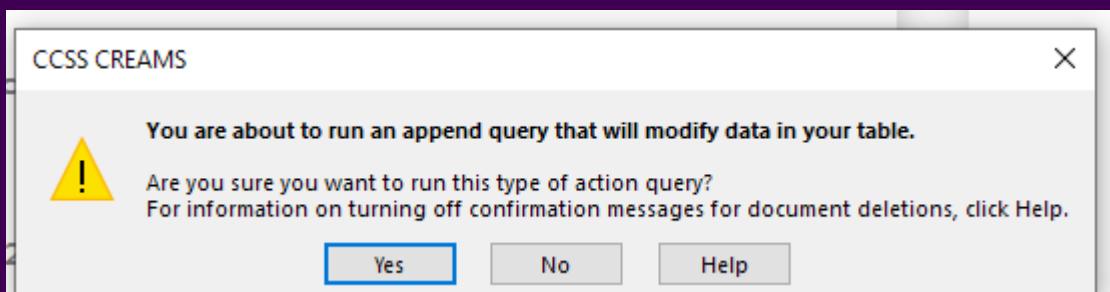


(The club field is 0 because there are no clubs in charge of this activity, if there is a club in charge it will show the club id.)

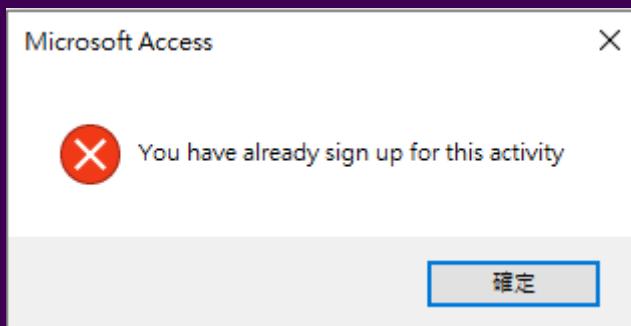
Student can sign up for this activity by pressing the "Sign Up" button at the bottom right



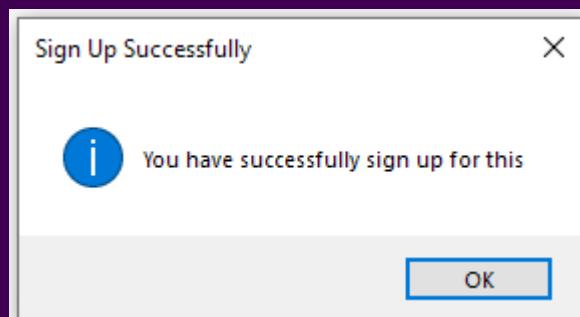
If student sign up a activity, it will modify the activity_record query



If they already signed up, this will pop up to remind them



If they sign up successfully, this message will pop up to inform the user.



VBA CODE

```
Option Compare Database

Private Sub btn_BackOverlay_Click()
    DoCmd.Close
End Sub

Private Sub btn_signUpOverlay_Click()
    If ((Nz(DCount("*", "Activity_record", "student_id = '" &
        getCurrentid & "' AND Activity_id=''" & getCurrentTMPid & "'"), 0)) <> 0) Then
        ErrorMessage (4)
    Else
        DoCmd.OpenQuery "Student_InsertRecords_Activity_record"
        If ((Nz(DCount("*", "Activity_record", "student_id = '" &
            getCurrentid & "' AND Activity_id=''" & getCurrentTMPid & "'"), 0)) = 0) Then
            ErrorMessage (99)
        End If
    End If
End Sub
```

```
        Else
            SuccessMessage (2)

        End If

    End If

End Sub

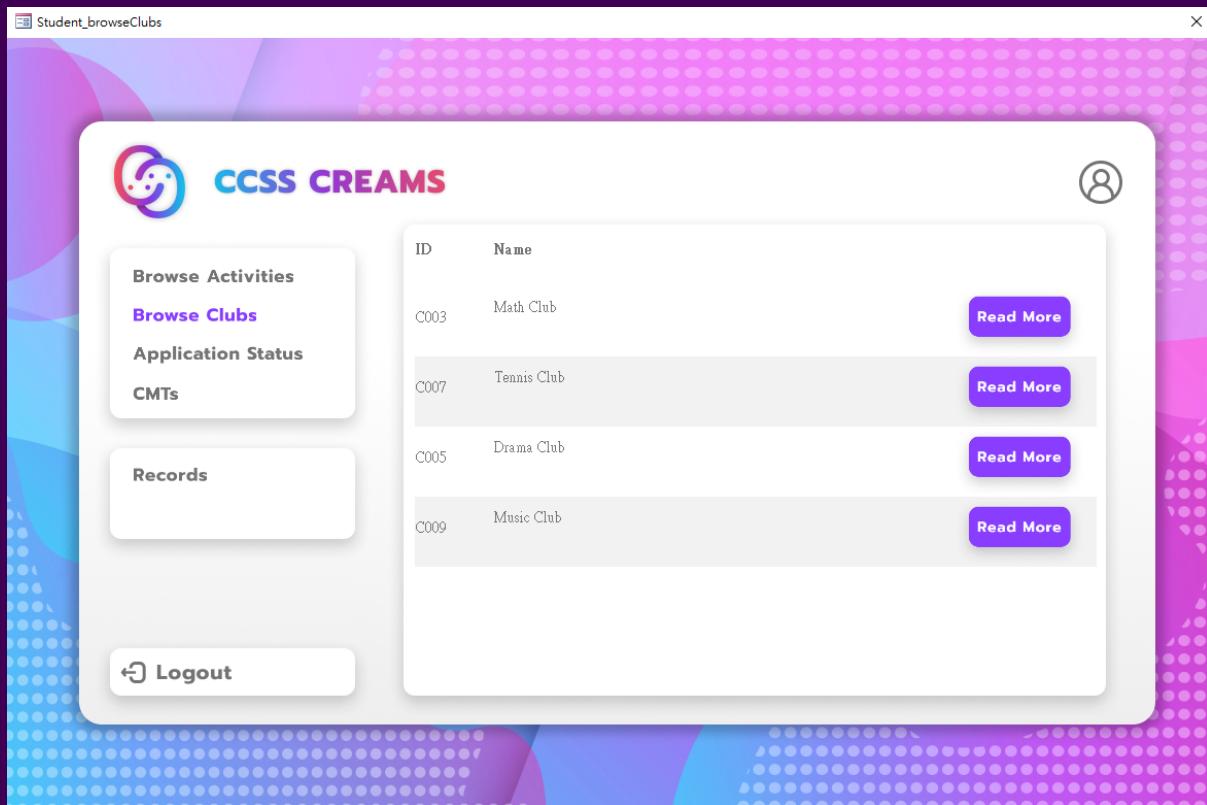
Private Sub Form_Load()

    Me.txt_nowDate.Caption = getNowDate
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_stdid.Caption = getCurrentid

    Me.act_id.Caption = getCurrentTMPid
    Me.act_name.Caption = Nz(DLookup("[Activity_name]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
    Me.act_type.Caption = Nz(DLookup("[Activity_type]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
    Me.act_part.Caption = Nz(DCount("Activity_id",
"Activity_record", "Activity_id = '" & getCurrentTMPid & "') ) &
"/" & Nz(DLookup("[Activity_maxParticipants]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
    Me.act_des.Caption = Nz(DLookup("[Activity_description]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
    Me.act_start.Caption = Nz(DLookup("[Activity_startDate]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0) & " " & Nz(DLookup("[Activity_startTime]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
    Me.act_end.Caption = Nz(DLookup("[Activity_endDate]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0) & " " & Nz(DLookup("[Activity_endTime]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
    Me.act_dead.Caption =
Nz(DLookup("[Activity_registrationDeadline]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0) & " " &
Nz(DLookup("[Activity_registrationDeadlineTime]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"') , 0)
```

```
Me.act_teach.Caption = Nz(DLookup("[Teacher_id]",  
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &  
"'"), 0)  
Me.act_club.Caption = Nz(DLookup("[Club_id]",  
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &  
"'"), 0)  
Me.act_loc.Caption = Nz(DLookup("[Activity_location]",  
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &  
"'"), 0)  
  
Dim penV As Integer  
Dim penD As String  
Dim appV As Integer  
  
penV = Nz(DLookup("[Activity_pending]",  
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &  
"'"), 0)  
  
If (penV = -1) Then  
    penD = "Yes"  
    Me.act_checkPen.Value = False  
  
Else  
    penD = "No"  
    Me.act_checkPen.Value = True  
  
End If  
  
Me.act_pen.Caption = penD  
End Sub
```

The next one is when students click browse club it will go to a new form which is Student_browseClubs form, the **available club that they can sign in will show up.**



VBA CODE

```
Option Compare Database

Private Sub btn_HomeLogoOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_HomeTextOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_LogoutOverlay_Click()

    ToPage "allLogin"
    setCurrentid ("")
End Sub
```

```
Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

Private Sub Form_Load()
    Me.txt_stdid.Caption = getCurrentid
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.txt_schoolyear.Caption = getCurrentSchoolYear

    Me.Refresh
    DoCmd.Requery
End Sub

Private Sub btn_BrowseActiviesOverlay_Click()
    ToTempPage "Student_browseActivities"
End Sub

Private Sub btn_BrowseClubsOverlay_Click()
    Me.Refresh
    DoCmd.Requery
End Sub

Private Sub btn_ApplicationStatusOverlay_Click()
    ToTempPage "Student_applicationStatus"
End Sub

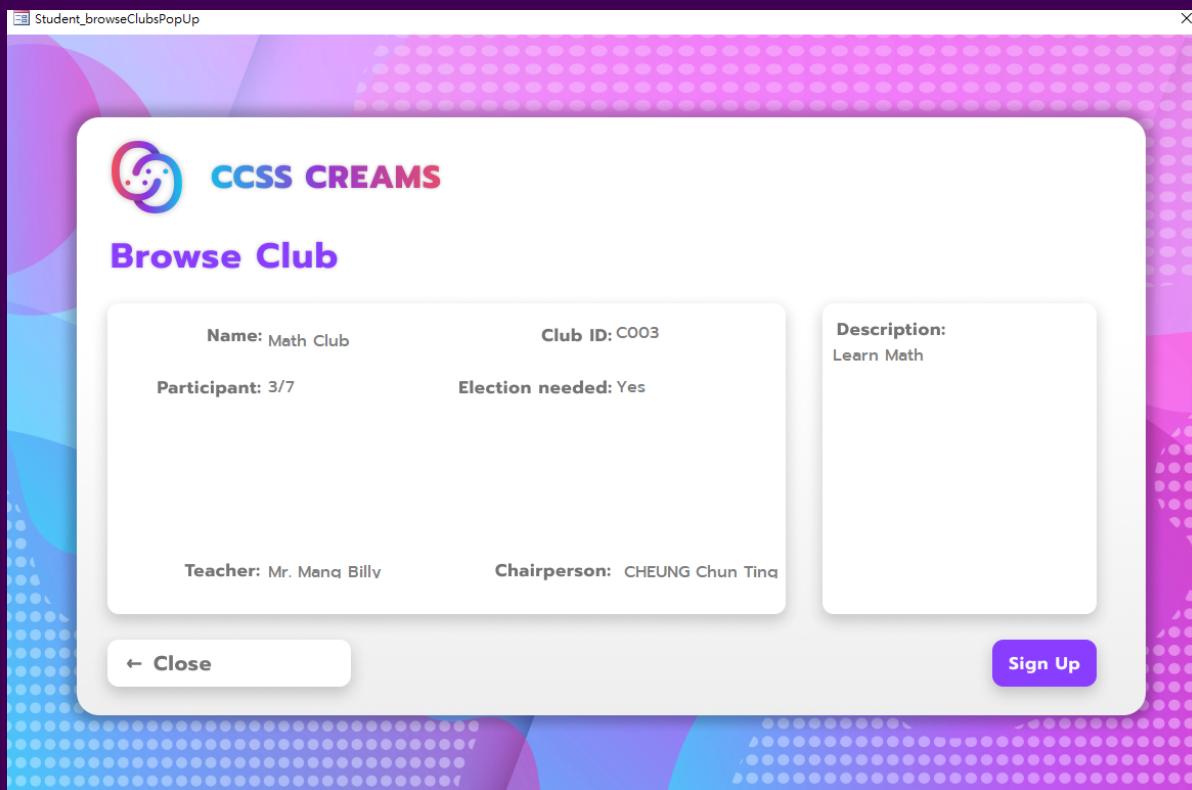
Private Sub btn_CMTsOverlay_Click()
    ToTempPage "Student_CMTs_popUP"
End Sub

Private Sub btn_recordsOverlay_Click()
```

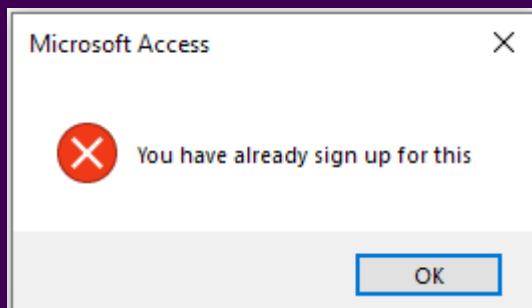
```
ToTempPage "Student_records"
```

```
End Sub
```

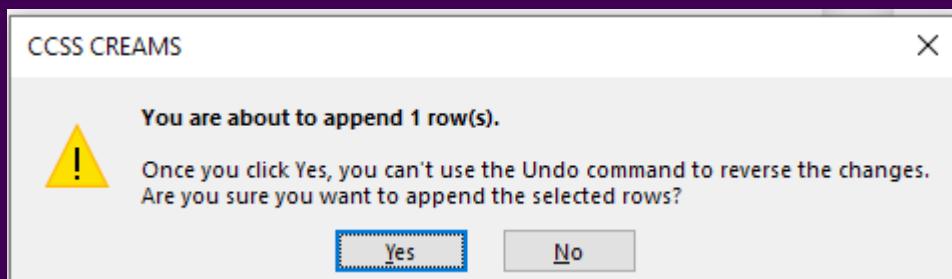
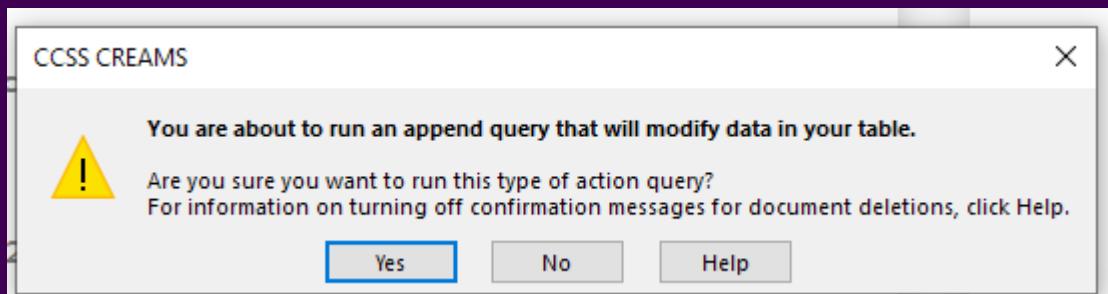
If a student is interested in one of the clubs they can click **read more** to know more about the club. Then it will lead them to another form which is Student_browseClubsPopup. And this will tell about **further details about the clubs**.



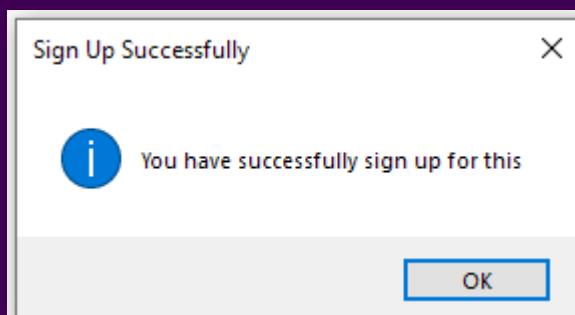
If they are still interested and consider becoming a member of that particular club, they can press Sign up just like joining an activity. If they already sign up error



If not yet joined then this will pops up



If they sign up successfully, this message will pop up to inform the user



VBA CODE

```
Option Compare Database

Private Sub btn_BackOverlay_Click()
    DoCmd.Close
End Sub

Private Sub btn_signUpOverlay_Click()
    If ((Nz(DCount("*", "Club_record", "Student_id = '" &
        getCurrentid & "' AND Club_id=''" & getCurrentTMPid & "'"), 0)) <>
        0) Then
        ErrorMessage (4)
    Else
        DoCmd.OpenQuery "Student_InsertRecords_Club_record"
    End If
End Sub
```

```
If ((Nz(DCount("*", "Club_record", "Student_id = '" &
getCurrentid & "' AND Club_id=''" & getCurrentTMPid & "'"), 0)) =
0) Then
    ErrorMessage (99)

Else
    SuccessMessage (2)

End If

End If

End Sub

Private Sub Form_Load()

Dim CurrentTmpGender As String
Dim TmpTeacherid As String
Dim TmpTeacherName As String
Dim TmpStudentid As String

Me.txt_nowDate.Caption = getNowDate
Me.txt_schoolyear.Caption = getCurrentSchoolYear
Me.txt_stdid.Caption = getCurrentid

Me.club_id.Caption = getCurrentTMPid
Me.Club_name.Caption = Nz(DLookup("[Club_name]",
"Club_information", "[Club_id] = '" & [getCurrentTMPid] & "'"),
0)
    Me.club_des.Caption = Nz(DLookup("[Club_description]",
"Club_information", "[Club_id] = '" & [getCurrentTMPid] & "'"),
0)
        Me.club_part.Caption = Nz(DCount("Club_id", "Club_record",
"Club_id = '" & getCurrentTMPid & "')") & "/" &
Nz(DLookup("[Club_maxParticipants]", "Club_schoolyear",
"[Club_id] = '" & [getCurrentTMPid] & "'"), 0)

TmpTeacherid = Nz(DLookup("[Teacher_id]", "Club_schoolyear",
"[Club_id] = '" & [getCurrentTMPid] & "'"), 0)
    CurrentTmpGender = Nz(DLookup("[Teacher_gender]",
"Teacher_information", "[Teacher_id] = '" & [TmpTeacherid] &
"'"), 0)

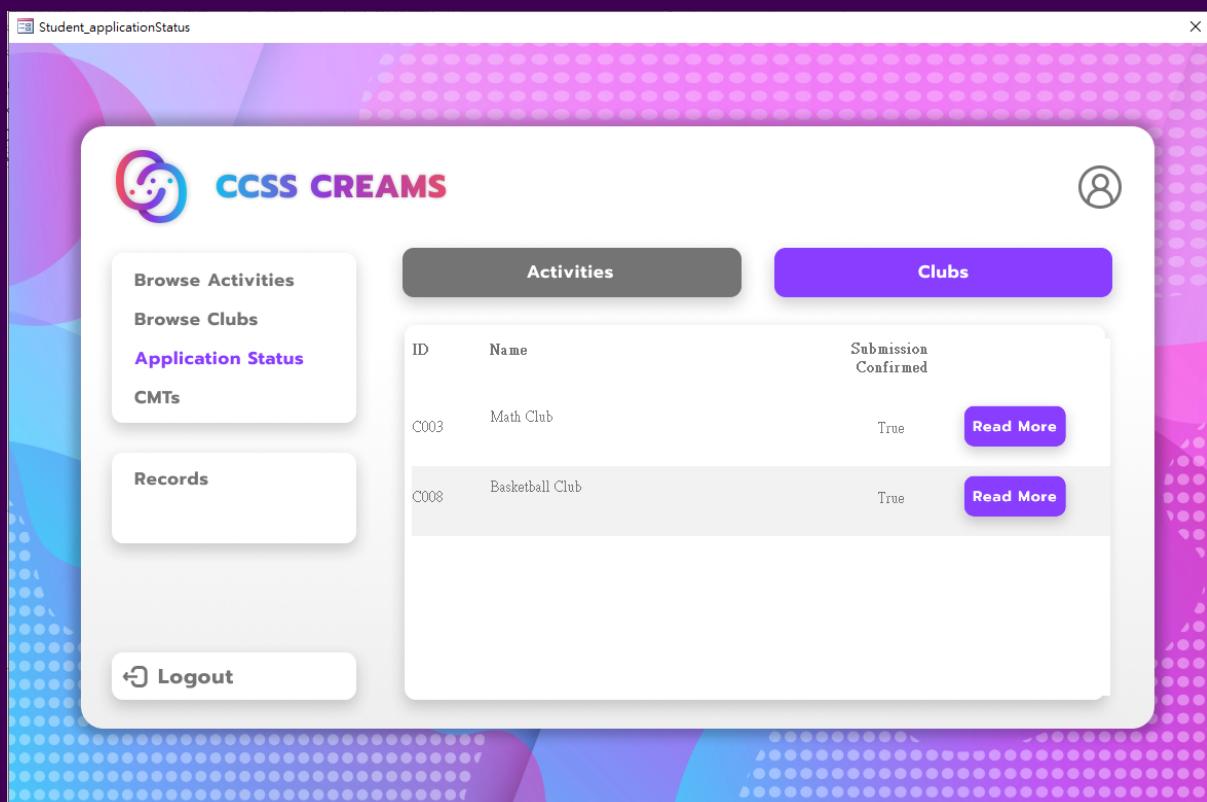
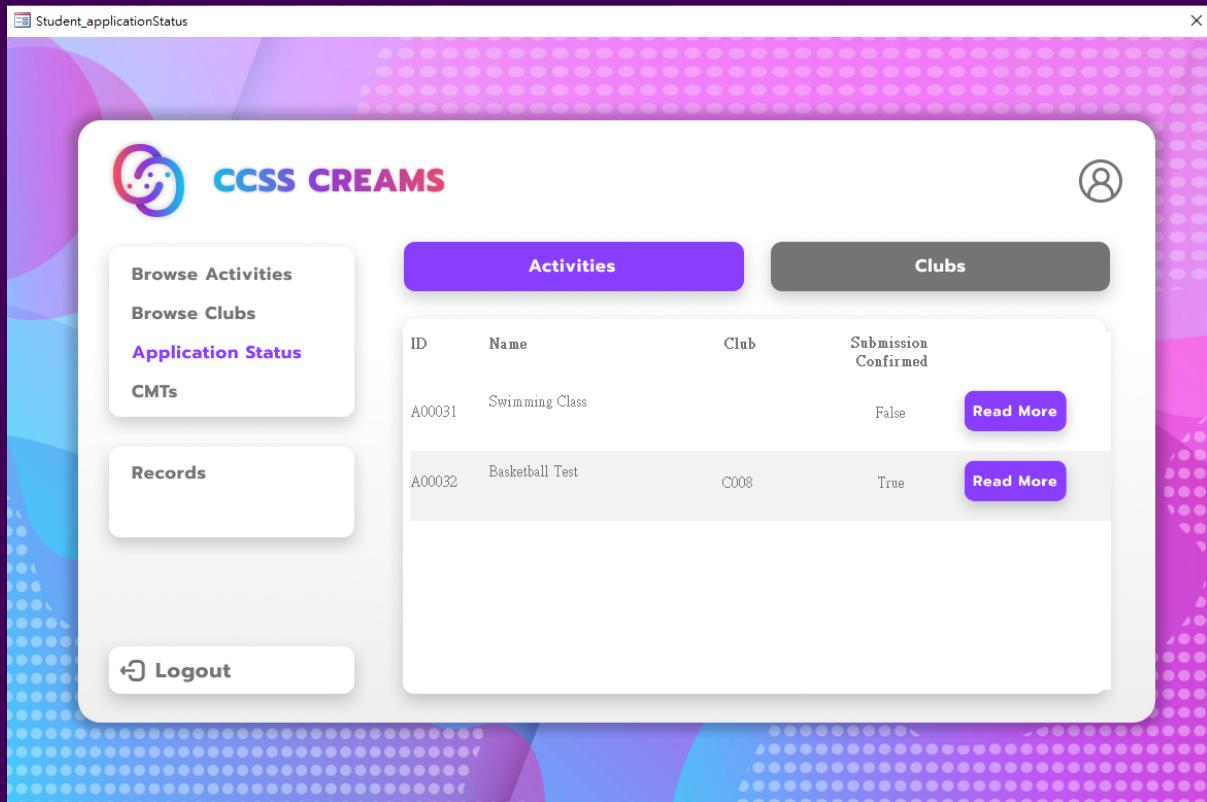
If CurrentTmpGender = "F" Then
    TmpTeacherName = "Ms."

Else
```

```
TmpTeacherName = "Mr."  
  
End If  
  
TmpTeacherName = TmpTeacherName & " " &  
Nz(DLookup("[Teacher_lastName]", "Teacher_information",  
"[Teacher_id] = '" & TmpTeacherid & "'"), 0) & " " &  
Nz(DLookup("[Teacher(firstName]", "Teacher_information",  
"[Teacher_id] = '" & TmpTeacherid & "'"), 0)  
Me.club_teach.Caption = TmpTeacherName  
  
TmpStudentid = Nz(DLookup("[Student_id]", "Club_schoolyear",  
"[Club_id] = '" & [getCurrentTMPid] & "'"), 0)  
Me.club_chairper.Caption = Nz(DLookup("[Student_lastName]",  
"Student_information", "[Student_id] = '" & TmpStudentid & "'"),  
0) & " " & Nz(DLookup("[Student(firstName]",  
"Student_information", "[Student_id] = '" & TmpStudentid & "'"),  
0)  
  
Dim penV As Integer  
Dim penD As String  
Dim appV As Integer  
  
penV = Nz(DLookup("[Club_pending]", "club_information",  
"[club_id] = '" & [getCurrentTMPid] & "'"), 0)  
  
If (penV = -1) Then  
penD = "Yes"  
Me.club_checkTpen.Value = False  
  
Else  
penD = "No"  
Me.club_checkTpen.Value = True  
  
End If  
  
Me.club_pen.Caption = penD  
  
End Sub
```

After browsing clubs they can check Application Status and that will bring them to another new form, which is **application status**, this will show the status that they sign up for.

They can choose to see "activities" or "clubs" status



VBA CODE

```
Option Compare Database

Private Sub btn_HomeLogoOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_HomeTextOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_LogoutOverlay_Click()

    ToPage "allLogin"
    setCurrentid ("")
    Me.btn_logout.Visible = False
    Me.btn_logoutOverlay.Visible = True

End Sub

Private Sub btn_longActOverlay_Click()

    Me.btn_longActP.Visible = True
    Me.btn_longAct.Visible = False
    Me.btn_longClubsP.Visible = False
    Me.btn_longClubs.Visible = True
    Me.Student_Querysubform_ApplicationStatus_act.Visible = True
    Me.Student_Querysubform_ApplicationStatus_club.Visible =
    False

    Me.Refresh
    Me.Requery
    Me.Form.Requery
    Me.Form.Refresh

End Sub

Private Sub btn_longClubsOverlay_Click()

    Me.btn_longActP.Visible = False
    Me.btn_longAct.Visible = True
    Me.Student_Querysubform_ApplicationStatus_club.Visible = True
    Me.Student_Querysubform_ApplicationStatus_act.Visible = False

End Sub
```

```
Me.btnAddClub.Visible = True
Me.btnAddClubs.Visible = False
Me.Student_Querysubform_ApplicationStatus_act.Visible = False
Me.Student_Querysubform_ApplicationStatus_club.Visible = True

Me.Refresh
Me.Requery
Me.Form.Requery
Me.Form.Refresh

End Sub

Private Sub btn_PersonOverlay_Click()

    ToPage "Student_Personal_general"

End Sub


Private Sub Form_Load()

    Me.txt_stdid.Caption = getCurrentid
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.txt_schoolyear.Caption = getCurrentSchoolYear

    Me.btnAddAct.Visible = True
    Me.btnAddAct.Visible = False
    Me.btnAddClub.Visible = False
    Me.btnAddClubs.Visible = True
    Me.Student_Querysubform_ApplicationStatus_act.Visible = True
    Me.Student_Querysubform_ApplicationStatus_club.Visible =
        False

    Me.Refresh
    Me.Requery
    Me.Form.Requery
    Me.Form.Refresh
    'Me.Student_Querysubform_ApplicationStatus_act.Form.Requery
    'Me.Student_Querysubform_ApplicationStatus_club.Form.Requery

End Sub

Private Sub btn_BrowseActivitiesOverlay_Click()

    ToTempPage "Student_browseActivities"

```

```
End Sub

Private Sub btn_BrowseClubsOverlay_Click()

    ToTempPage "Student_browseClubs"

End Sub

Private Sub btn_ApplicationStatusOverlay_Click()

    Me.Refresh
    DoCmd.Requery

End Sub

Private Sub btn_CMTsOverlay_Click()

    ToTempPage "Student_CMTs_popUP"

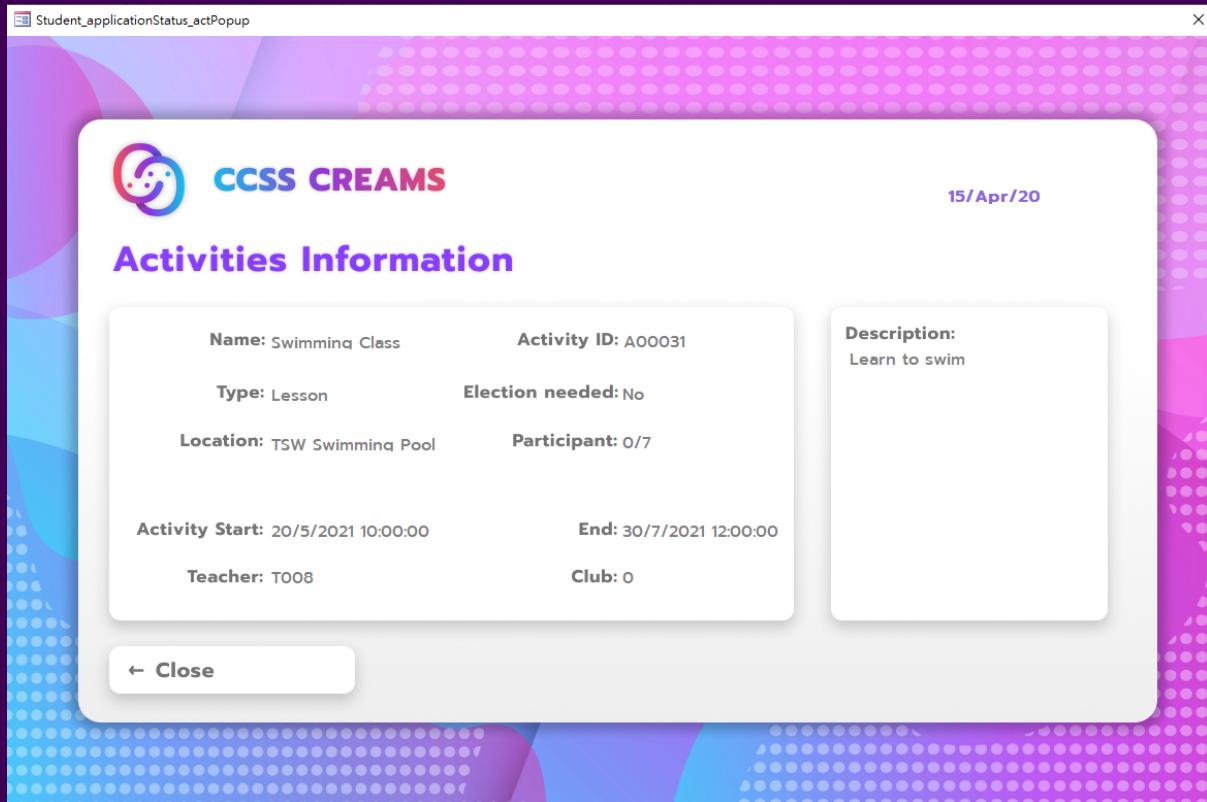
End Sub

Private Sub btn_recordsOverlay_Click()

    ToTempPage "Student_records"

End Sub
```

When they want to **check the information of the activity**, they can click read more and this will bring them to student_applicationStatus_actPopup.



As this time, they already signed up so the sign up button disappeared.

VBA CODE

```
Option Compare Database

Private Sub btn_BackOverlay_Click()
    DoCmd.Close
End Sub

Private Sub Form_Load()
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_stdid.Caption = getCurrentid

    Me.act_id.Caption = getCurrentTMPid

```

```

Me.act_name.Caption = Nz(DLookup("[Activity_name]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
    Me.act_type.Caption = Nz(DLookup("[Activity_type]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
        Me.act_part.Caption = Nz(DCount("Activity_id",
"Activity_record", "Activity_id = '" & getCurrentTMPid & "' AND
Activity_approval = TRUE)) & "/" &
Nz(DLookup("[Activity_maxParticipants]", "Activity_information",
"[Activity_id] = '" & [getCurrentTMPid] & "'"), 0)
    Me.act_des.Caption = Nz(DLookup("[Activity_description]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
        Me.act_start.Caption = Nz(DLookup("[Activity_startDate]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0) & " " & Nz(DLookup("[Activity_startTime]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
    Me.act_end.Caption = Nz(DLookup("[Activity_endDate]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0) & " " & Nz(DLookup("[Activity_endTime]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
    Me.act_teach.Caption = Nz(DLookup("[Teacher_id]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
        Me.act_club.Caption = Nz(DLookup("[Club_id]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)
    Me.act_loc.Caption = Nz(DLookup("[Activity_location]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)

Dim penV As Integer
Dim penD As String
Dim appV As Integer

penV = Nz(DLookup("[Activity_pending]",
"Activity_information", "[Activity_id] = '" & [getCurrentTMPid] &
"'"), 0)

If (penV = -1) Then
    penD = "Yes"
    Me.act_checkPen.Value = False

```

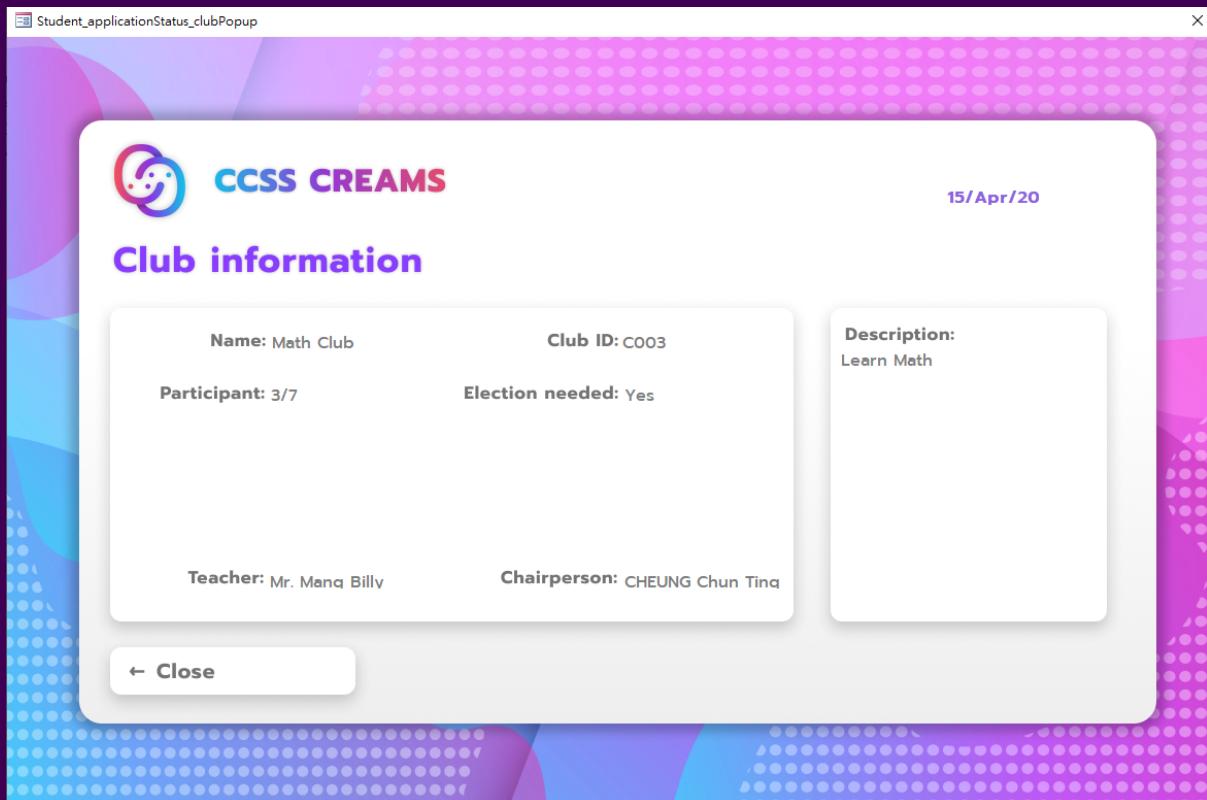
```
Else
    penD = "No"
    Me.act_checkTpen.Value = True

End If

Me.act_pen.Caption = penD

End Sub
```

When they want to **check** the **information of the club**, they can click **read more** and this will bring them to student_applicationStatus_clubPopup.



As this time, they already signed up so the sign up button disappeared.

VBA CODE

```
Option Compare Database

Private Sub btn_BackOverlay_Click()
    DoCmd.Close
End Sub

Private Sub Form_Load()
    Dim CurrentTmpGender As String
    Dim TmpTeacherid As String
    Dim TmpTeacherName As String
    Dim TmpStudentid As String

    Me.txt_nowDate.Caption = getNowDate

```

```
Me.txt_schoolyear.Caption = getCurrentSchoolYear
Me.txt_stdid.Caption = getCurrentid

Me.club_id.Caption = getCurrentTMPid
Me.Club_name.Caption = Nz(DLookup("[Club_name]",
"Club_information", "[Club_id] = '" & [getCurrentTMPid] & "'"),
0)
Me.club_des.Caption = Nz(DLookup("[Club_description]",
"Club_information", "[Club_id] = '" & [getCurrentTMPid] & "'"),
0)
Me.club_part.Caption = Nz(DCount("Club_id", "Club_record",
"Club_id = '" & getCurrentTMPid & "')") & "/" &
Nz(DLookup("[Club_maxParticipants]", "Club_schoolyear",
"[Club_id] = '" & [getCurrentTMPid] & "'"), 0)

TmpTeacherid = Nz(DLookup("[Teacher_id]", "Club_schoolyear",
"[Club_id] = '" & [getCurrentTMPid] & "'"), 0)
CurrentTmpGender = Nz(DLookup("[Teacher_gender]",
"Teacher_information", "[Teacher_id] = '" & [TmpTeacherid] &
"'"), 0)

If CurrentTmpGender = "F" Then
    TmpTeacherName = "Ms."
Else
    TmpTeacherName = "Mr."
End If

TmpTeacherName = TmpTeacherName & " " &
Nz(DLookup("[Teacher_lastName]", "Teacher_information",
"[Teacher_id] = '" & TmpTeacherid & "'"), 0) & " " &
Nz(DLookup("[Teacher(firstName]", "Teacher_information",
"[Teacher_id] = '" & TmpTeacherid & "'"), 0)
Me.club_teach.Caption = TmpTeacherName

TmpStudentid = Nz(DLookup("[Student_id]", "Club_schoolyear",
"[Club_id] = '" & [getCurrentTMPid] & "'"), 0)
Me.club_chairper.Caption = Nz(DLookup("[Student_lastName]",
"Student_information", "[Student_id] = '" & TmpStudentid & "'"),
0) & " " & Nz(DLookup("[Student(firstName]",
"Student_information", "[Student_id] = '" & TmpStudentid & "'"),
0)

Dim penV As Integer
Dim penD As String
Dim appV As Integer
```

```
penV = Nz(DLookup("[Club_pending]", "club_information",
"[club_id] = '" & [getCurrentTMPid] & "'"), 0)

If (penV = -1) Then
pend = "Yes"
Me.club_checkTpen.Value = False

Else
pend = "No"
Me.club_checkTpen.Value = True

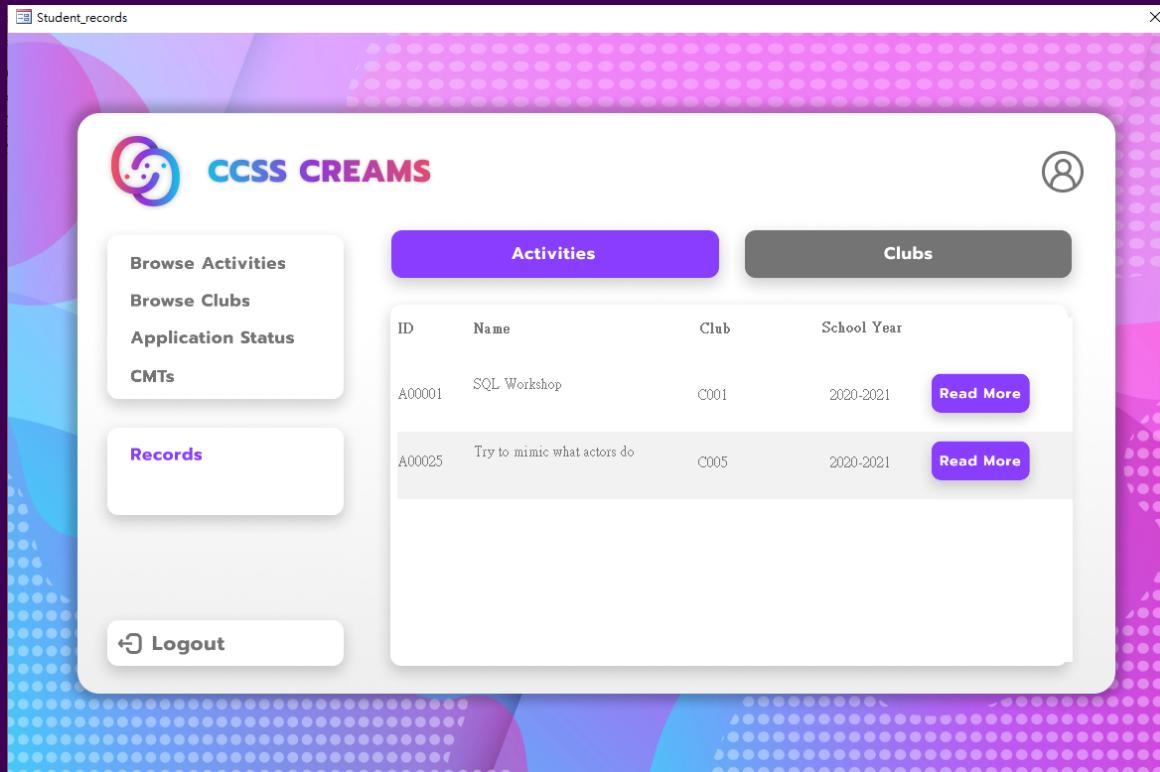
End If

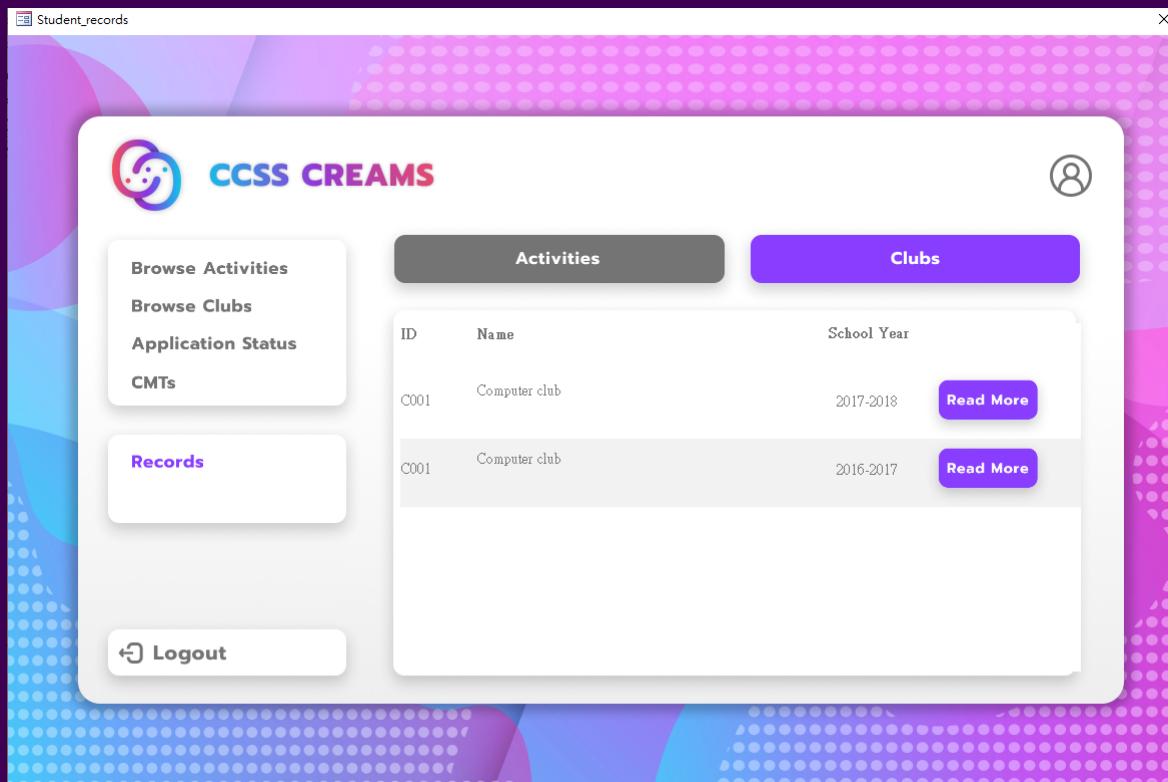
Me.club_pen.Caption = pend

End Sub
```

As CMTs will bring to another form that is completely different so now, we skip it first.

When a student clicks on the records button, it will bring them to a new form that is the Student_record form and this will display the **activity and club participated record** and they can switch to check activity and club record by clicking the activities and club button. This will store **cross year records** so students can refer to it even if the activity is held a long time ago.





VBA CODE

```
Option Compare Database

Private Sub btn_HomeLogoOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_HomeTextOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_LogoutOverlay_Click()

    ToPage "allLogin"
    setCurrentid ("")
End Sub

Private Sub btn_longActOverlay_Click()
```

```
Me.btnAddP.Visible = True
Me.btnAdd.Visible = False
Me.btnAddClubsP.Visible = False
Me.btnAddClubs.Visible = True
Me.Student_Querysubform_Records_act.Visible = True
Me.Student_Querysubform_Records_club.Visible = False

Me.Refresh
Me.Requery
Me.Form.Requery
Me.Form.Refresh

End Sub

Private Sub btn_longClubsOverlay_Click()

Me.btnAddP.Visible = False
Me.btnAdd.Visible = True
Me.btnAddClubsP.Visible = True
Me.btnAddClubs.Visible = False
Me.Student_Querysubform_Records_club.Visible = True
Me.Student_Querysubform_Records_act.Visible = False

Me.Refresh
Me.Requery
Me.Form.Requery
Me.Form.Refresh

End Sub

Private Sub btn_PersonOverlay_Click()

ToPage "Student_Personal_general"

End Sub

Private Sub Form_Load()

Me.txt_stdid.Caption = getCurrentid
Me.txt_nowDate.Caption = getNowDate
Me.txt_nowTime.Caption = getNowTime
Me.txt_schoolyear.Caption = getCurrentSchoolYear

Me.btnAddP.Visible = True
Me.btnAdd.Visible = False
```

```
Me.btnAddClubSP.Visible = False
Me.btnAddClub.Visible = True
Me.Student_Querysubform_Records_club.Visible = False
Me.Student_Querysubform_Records_act.Visible = True

Me.Refresh
Me.Requery
Me.Form.Requery
Me.Form.Refresh

End Sub

Private Sub btnBrowseActivitiesOverlay_Click()
    ToTempPage "Student_browseActivities"
End Sub

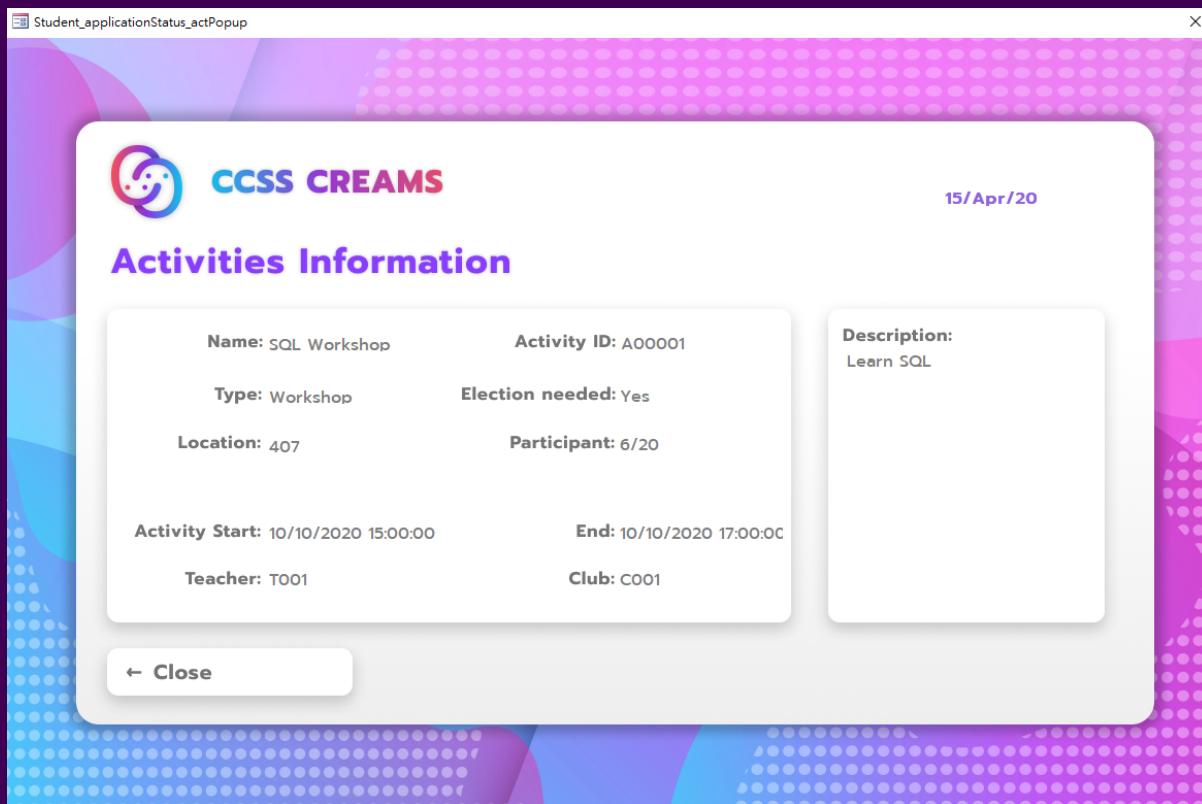
Private Sub btnBrowseClubsOverlay_Click()
    ToTempPage "Student_browseClubs"
End Sub

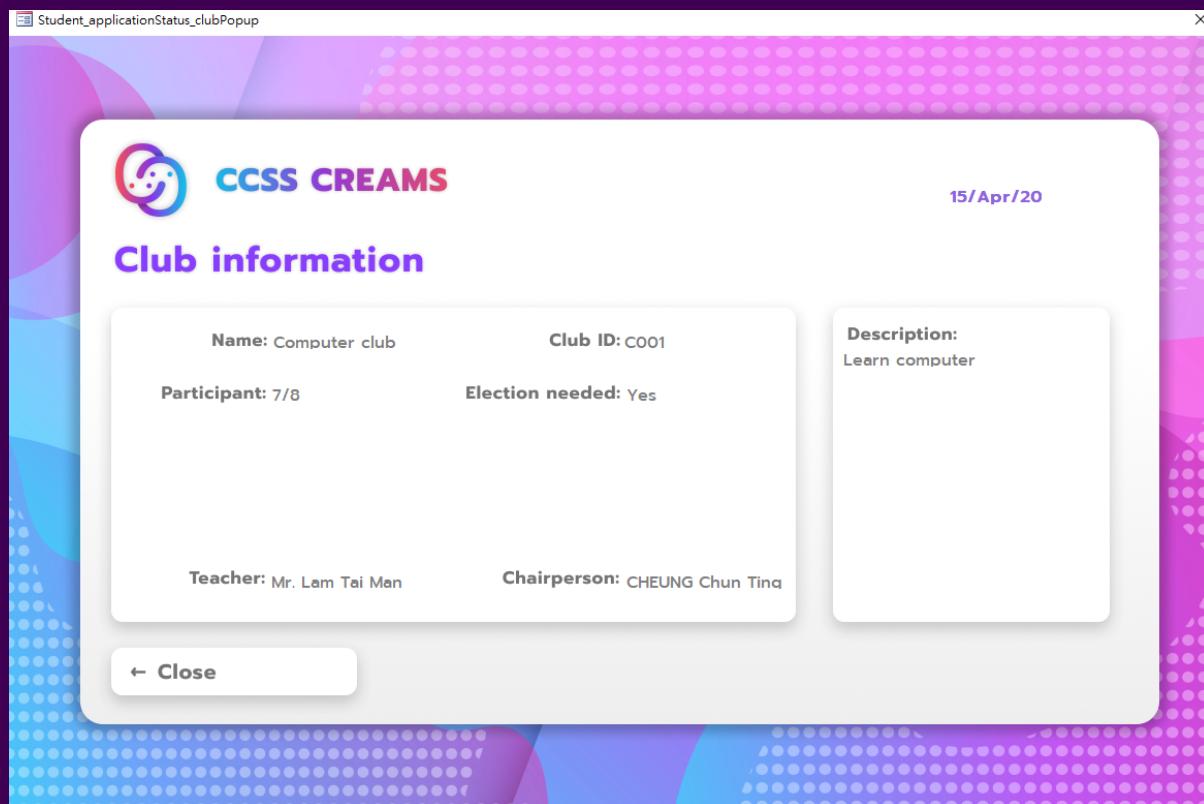
Private Sub btnApplicationStatusOverlay_Click()
    ToTempPage "Student_applicationStatus"
End Sub

Private Sub btnCMTsOverlay_Click()
    ToTempPage "Student_CMTs_popUP"
End Sub

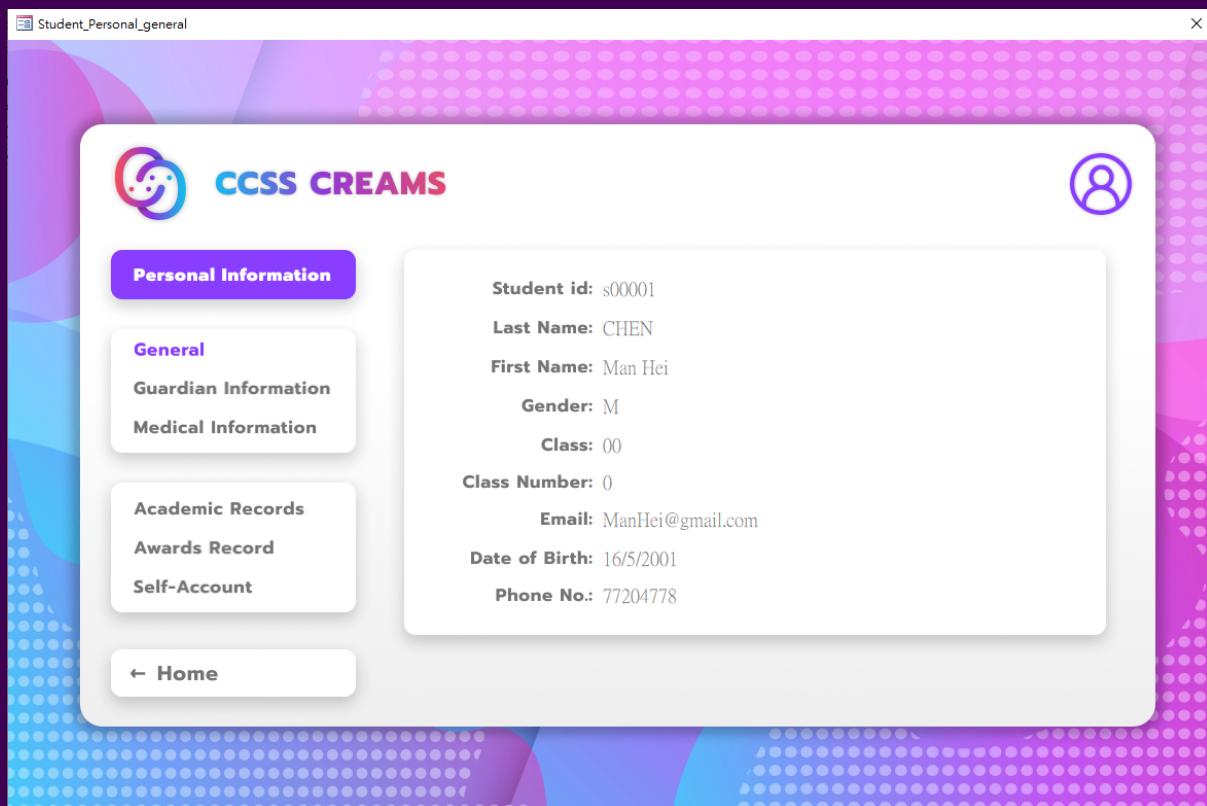
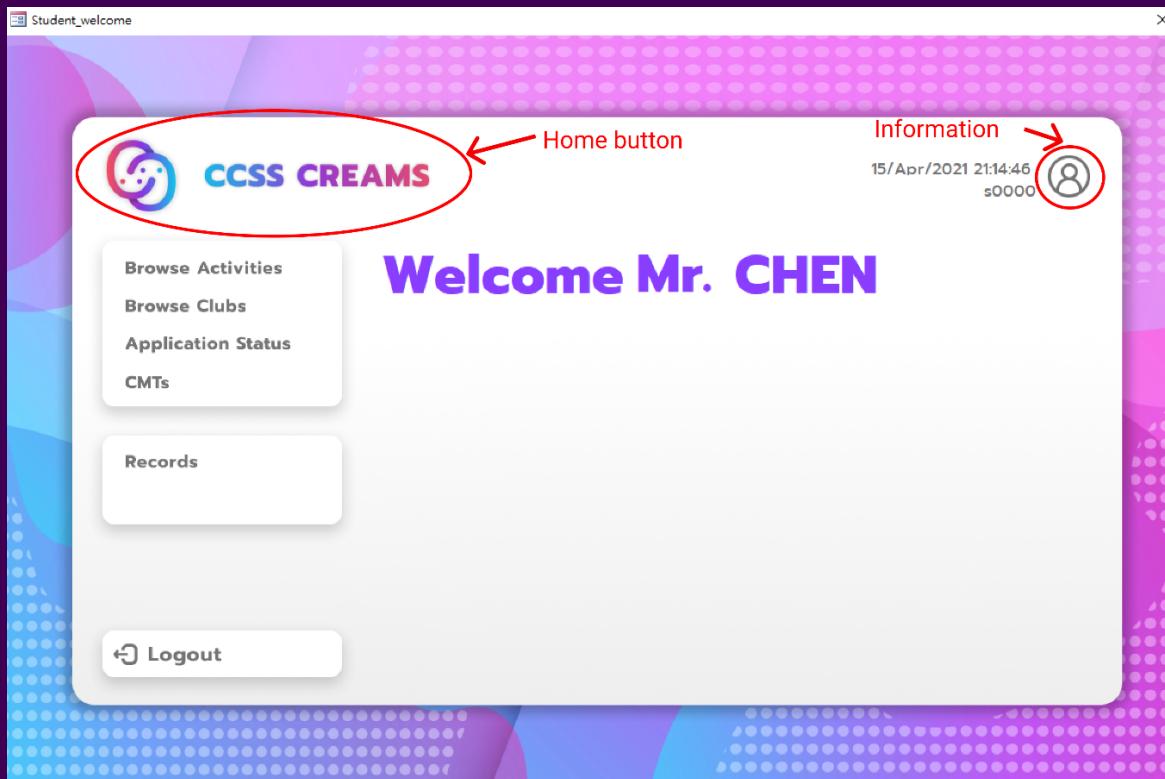
Private Sub btnrecordsOverlay_Click()
    Me.Refresh
    DoCmd.Requery
End Sub
```

As like the Student_applicationStatus, there will be a Read More button and this can let students check out the **further detailed information** of each **club** and **activities**. As the activity is already expired and the club record is last year's and even before that so there are no sign up buttons also. As it is displaying the same thing so we use the Student_applicationStatus_actPopup and Student_applicationStatus_clubPopup form to do this

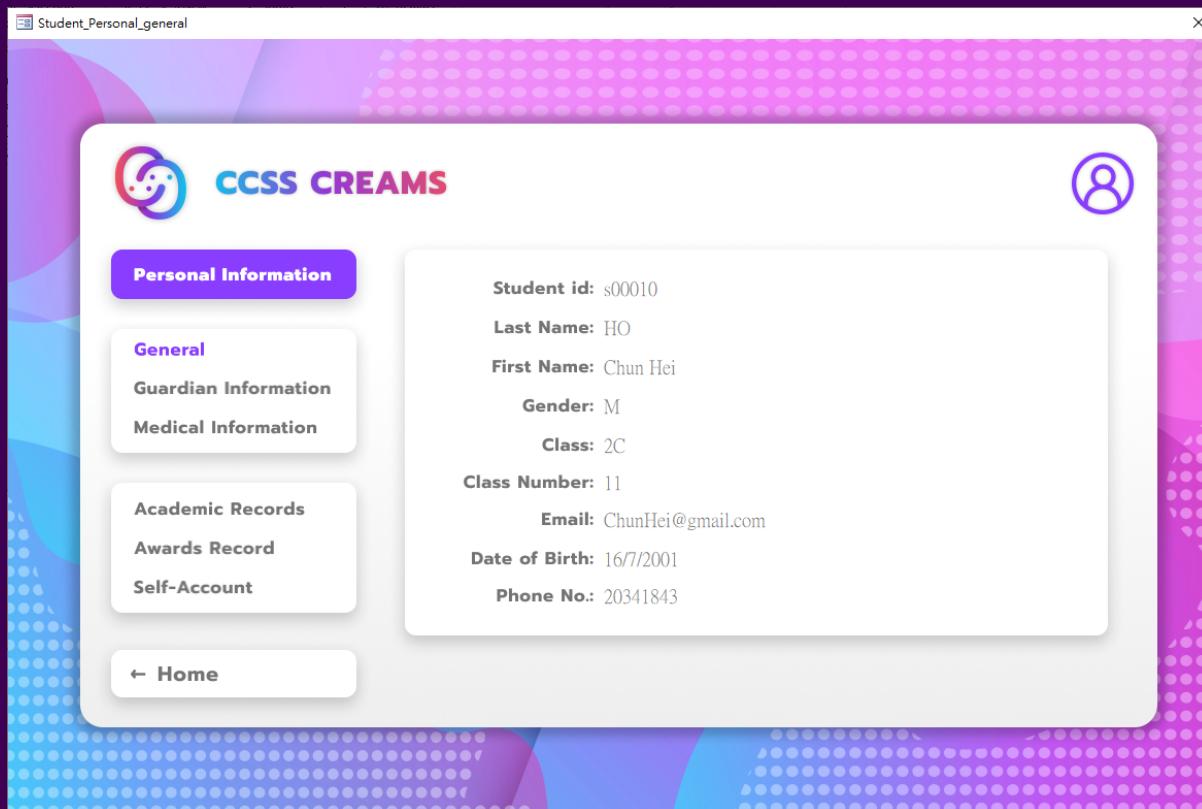




The Last one before CMTs is when clicking the circle in the top right corner with a cartoon human there, we can go to the student_Personal_general page and this will take to a new left sidebar which contains students can click to and check their **personal information** by themselves.



You can see in the class and **class number is 0** is because this student is a **graduated** student. We check a **not yet graduated** student and this is the result.



VBA CODE

```
Option Compare Database
Option Explicit

Private Sub btn_AcademicRecordsOverlay_Click()
    ToPage "Student_Personal_academicRecords"
End Sub

Private Sub btn_AwardsRecordOverlay_Click()
    ToPage "Student_Personal_awardsRecord"
End Sub

Private Sub btn_GeneralOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

Private Sub btn_GuardianInformationOverlay_Click()
```

```
ToPage "Student_Personal_guardianInformation"

End Sub

Private Sub btn_HomeOverlay_Click()

    BackToStudentHome

End Sub

Private Sub btn_MedicalInformationOverlay_Click()

    ToPage "Student_Personal_medicalInformation"

End Sub

Private Sub btn_PersonOverlay_Click()

    ToPage "Student_Personal_general"

End Sub

Private Sub btn_SelfAccountOverlay_Click()

    ToPage "Student_Personal_selfAccount"

End Sub

Private Sub Form_Load()

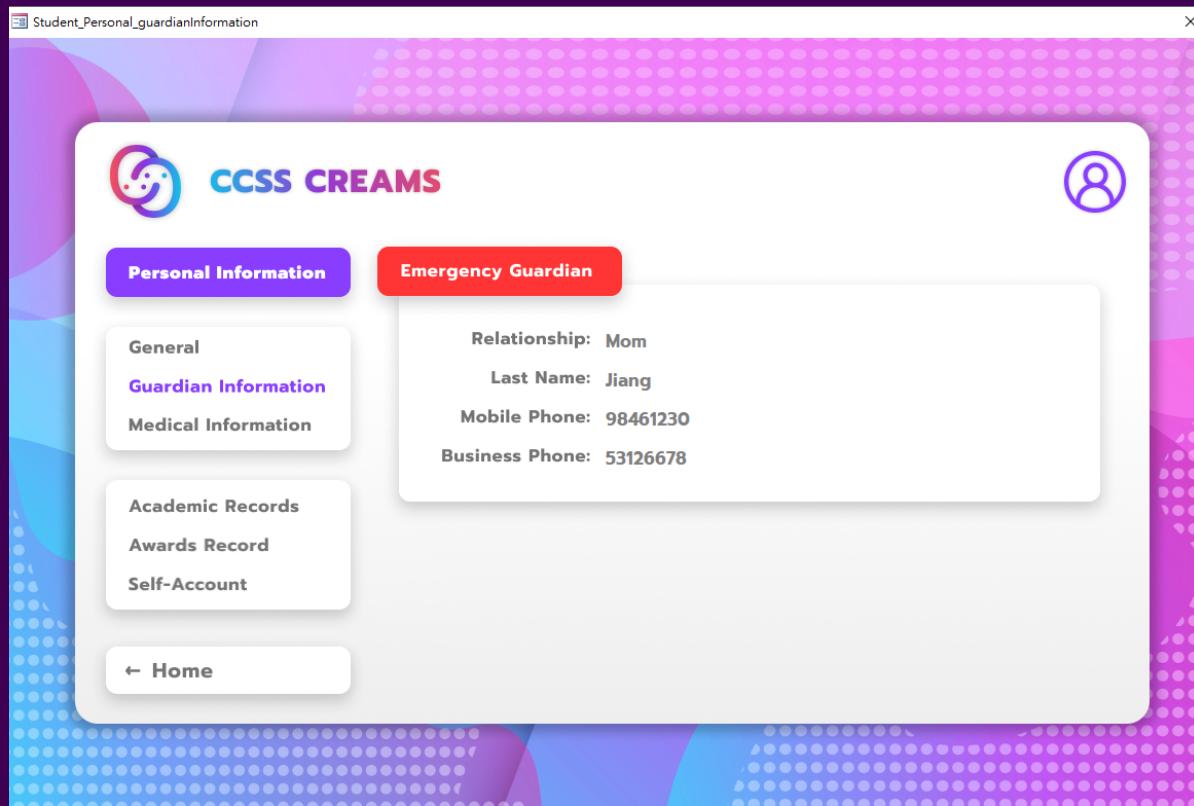
    Me.Student_id.Caption = getCurrentid
    Me.Student_Lname.Caption = Nz(DLookup("[Student_lastName]", "Student_information", "[Student_id] = '" & [getCurrentid] & "'", 0))
    Me.Student_Fname.Caption = Nz(DLookup("[Student(firstName]", "Student_information", "[Student_id] = '" & [getCurrentid] & "'", 0))

    Me.Student_gender.Caption = Nz(DLookup("[Student_gender]", "Student_information", "[Student_id] = '" & [getCurrentid] & "'", 0))
    Me.Student_class.Caption = Nz(DLookup("[Student_grade]", "Student_grade", "[Student_id] = '" & [getCurrentid] & "' AND [SchoolYear] = '" & [getCurrentSchoolYear] & "'", 0) &
        Nz(DLookup("[Student_class]", "Student_grade", "[Student_id] = '"
```

```
& [getCurrentid] & "' AND [SchoolYear] = '" &
[getCurrentSchoolYear] & "''), 0)
Me.Student_classnumber.Caption =
Nz(DLookup("[Student_classNumber]", "Student_grade",
"[Student_id] = '" & [getCurrentid] & "' AND [SchoolYear] = '" &
[getCurrentSchoolYear] & "''), 0)

Me.Student_email.Caption = Nz(DLookup("[Student_email]",
"Student_information", "[Student_id] = '" & [getCurrentid] &
"''), 0)
Me.Student_dob.Caption = Nz(DLookup("[Student_dob]",
"Student_information", "[Student_id] = '" & [getCurrentid] &
"''), 0)
Me.Student_phoneNumber.Caption =
Nz(DLookup("[Student_phoneNumber]", "Student_information",
"[Student_id] = '" & [getCurrentid] & "''), 0)
End Sub
```

If we click on the Guardian information, it will bring the Student_Personal_guardianInformation form and the form shows the **latest version of the guardian information** in the Student_guardian table.



VBA of Student_Personal_guardianInformation

```
Option Compare Database
```

```
Option Explicit
```

```
Private Sub btn_AcademicRecordsOverlay_Click()
```

```
    ToPage "Student_Personal_academicRecords"
```

```
End Sub
```

```
Private Sub btn_AwardsRecordOverlay_Click()
```

```
    ToPage "Student_Personal_awardsRecord"
```

```
End Sub
```

```
Private Sub btn_GeneralOverlay_Click()
```

```
    ToPage "Student_Personal_general"
```

```
End Sub
```

```
Private Sub btn_GuardianInformationOverlay_Click()
    ToPage "Student_Personal_guardianInformation"
End Sub

Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

Private Sub btn_MedicalInformationOverlay_Click()
    ToPage "Student_Personal_medicalInformation"
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

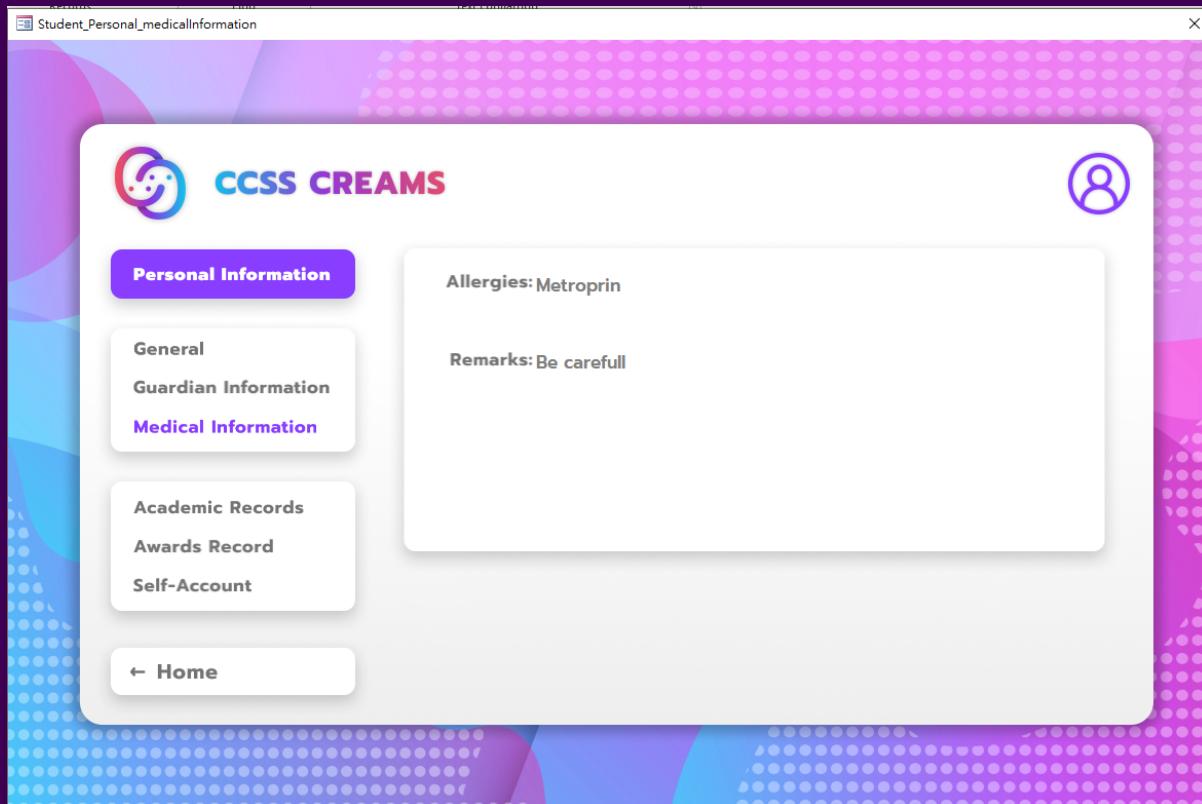
Private Sub btn_SelfAccountOverlay_Click()
    ToPage "Student_Personal_selfAccount"
End Sub

Function getGuardianMaxVersion() As Integer
    getGuardianMaxVersion = Nz(DMax("[Student_Guardian_version]", "Student_guardian", "[Student_id] = '" & [getCurrentid] & "'"), 0)
End Function

Public Sub Form_Load()
    Me.std_guarRealtsip.Caption =
    Nz(DLookup("[Student_Guardian_relationship]", "Student_guardian", "[Student_id] = '" & [getCurrentid] & "' AND
    [Student_Guardian_version] = " & getGuardianMaxVersion), 0)
    Me.std_guarLname.Caption =
    Nz(DLookup("[Student_Guardian_lastname]", "Student_guardian",
```

```
"[Student_id] = '" & [getCurrentid] & "' AND  
[Student_Guardian_version] = " & getGuardianMaxVersion), 0)  
    Me.std_guarMobile.Caption =  
    Nz(DLookup("[Student_Guardian_mobilePhoneNumber]",  
"Student_guardian", "[Student_id] = '" & [getCurrentid] & "' AND  
[Student_Guardian_version] = " & getGuardianMaxVersion), 0)  
    Me.std_guarBusiness.Caption =  
    Nz(DLookup("[Student_Guardian_businessPhoneNumber]",  
"Student_guardian", "[Student_id] = '" & [getCurrentid] & "' AND  
[Student_Guardian_version] = " & getGuardianMaxVersion), 0)  
  
End Sub
```

The next is when the student clicks Medical Information, it will go to the Student_Personal_medicalInformation form and the form shows what **medical allergy** the student has.



VBA of Student_Personal_medicalInformation

```
Option Compare Database
```

```
Option Explicit
```

```
Private Sub btn_AcademicRecordsOverlay_Click()
```

```
    ToPage "Student_Personal_academicRecords"
```

```
End Sub
```

```
Private Sub btn_AwardsRecordOverlay_Click()
```

```
    ToPage "Student_Personal_awardsRecord"
```

```
End Sub
```

```
Private Sub btn_GeneralOverlay_Click()
```

```
    ToPage "Student_Personal_general"
```

```
End Sub

Private Sub btn_GuardianInformationOverlay_Click()
    ToPage "Student_Personal_guardianInformation"
End Sub

Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

Private Sub btn_MedicalInformationOverlay_Click()
    ToPage "Student_Personal_medicalInformation"
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

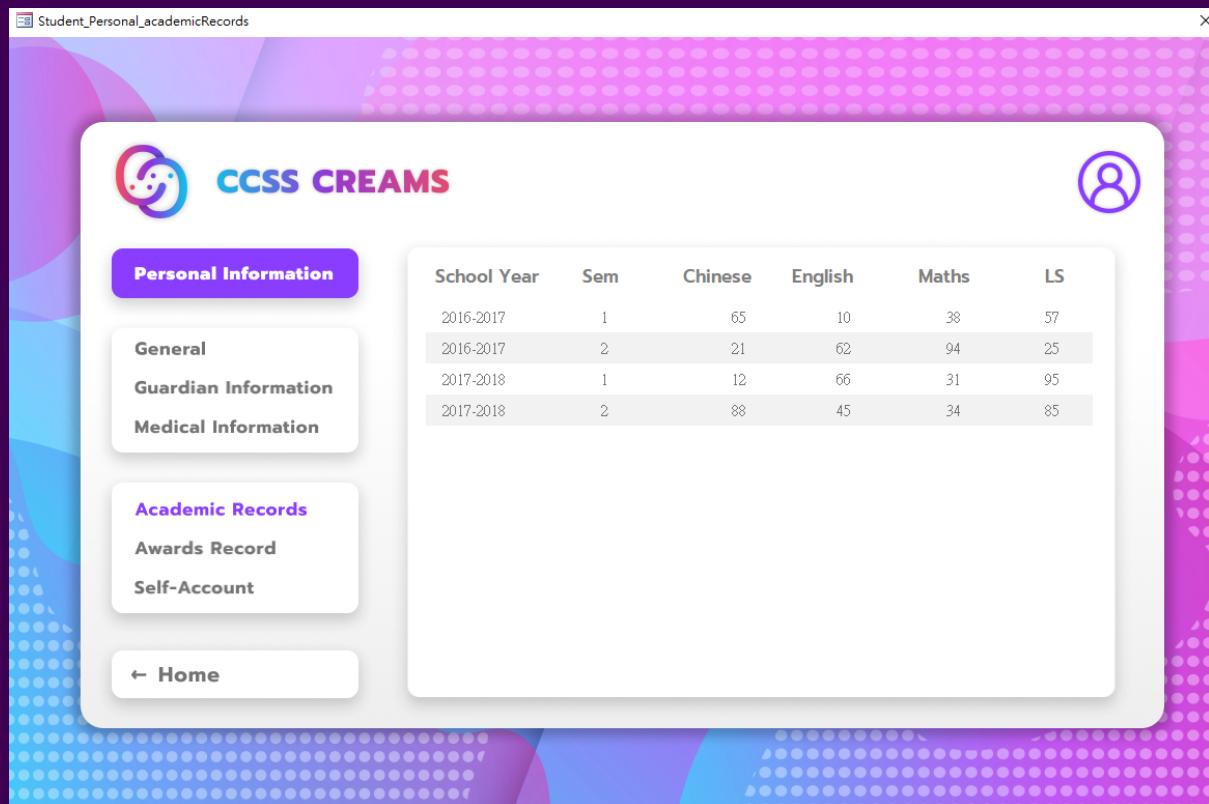
Private Sub btn_SelfAccountOverlay_Click()
    ToPage "Student_Personal_selfAccount"
End Sub

Function getMedicalMaxVersion() As Integer
    getMedicalMaxVersion = Nz(DMax("[Student_Medical_version]",
    "Student_medical", "[Student_id] = '" & [getCurrentid] & "'"), 0)
End Function

Public Sub Form_Load()
    Me.std_medAllergies.Caption =
    Nz(DLookup("[Student_Medical_allergies]", "Student_medical",
    "[Student_id] = '" & [getCurrentid] & "' AND
    [Student_Medical_version] = " & getMedicalMaxVersion), 0)
```

```
Me.std_medRemarks.Caption =  
Nz(DLookup("[Student_Medical_remarks]", "Student_medical",  
"[Student_id] = '" & [getCurrentid] & "' AND  
[Student_Medical_version] = " & getMedicalMaxVersion), 0)  
  
End Sub
```

The next one in the list is Academic Result which brings students to the Student_Personal_academicRecord form and it shows **all the academic records** of students in **every year every semester**.



VBA of Student_Personal_academicRecord

```
Option Compare Database
```

```
Option Explicit
```

```
Private Sub btn_AcademicRecordsOverlay_Click()
```

```
    ToPage "Student_Personal_academicRecords"
```

```
End Sub
```

```
Private Sub btn_AwardsRecordOverlay_Click()
```

```
    ToPage "Student_Personal_awardsRecord"
```

```
End Sub
```

```
Private Sub btn_GeneralOverlay_Click()
```

```
    ToPage "Student_Personal_general"
```

```
End Sub

Private Sub btn_GuardianInformationOverlay_Click()
    ToPage "Student_Personal_guardianInformation"
End Sub

Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

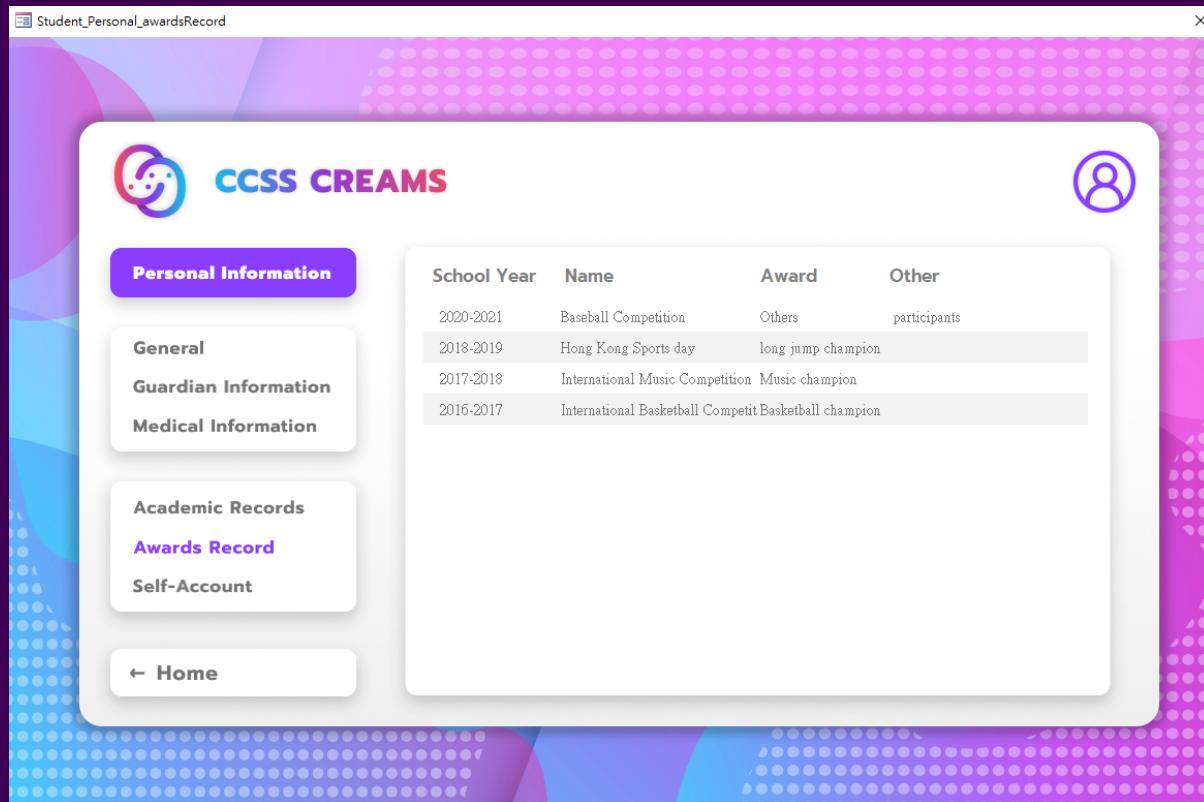
Private Sub btn_MedicalInformationOverlay_Click()
    ToPage "Student_Personal_medicalInformation"
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

Private Sub btn_SelfAccountOverlay_Click()
    ToPage "Student_Personal_selfAccount"
End Sub

Private Sub Form_Load()
    Me.txt_stdid.Caption = getCurrentid
    DoCmd.Requery
End Sub
```

After that when clicking the Awards button, it will go to the Student_Personal_awardsRecord form and it will display **all the awards** that the student has **won** and the year of it.



VBA of Student_Personal_awardsRecord

```
Option Compare Database
```

```
Option Explicit
```

```
Private Sub btn_AcademicRecordsOverlay_Click()
```

```
    ToPage "Student_Personal_academicRecords"
```

```
End Sub
```

```
Private Sub btn_AwardsRecordOverlay_Click()
```

```
    ToPage "Student_Personal_awardsRecord"
```

```
End Sub
```

```
Private Sub btn_GeneralOverlay_Click()
```

```
    ToPage "Student_Personal_general"
```

```
End Sub

Private Sub btn_GuardianInformationOverlay_Click()
    ToPage "Student_Personal_guardianInformation"
End Sub

Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

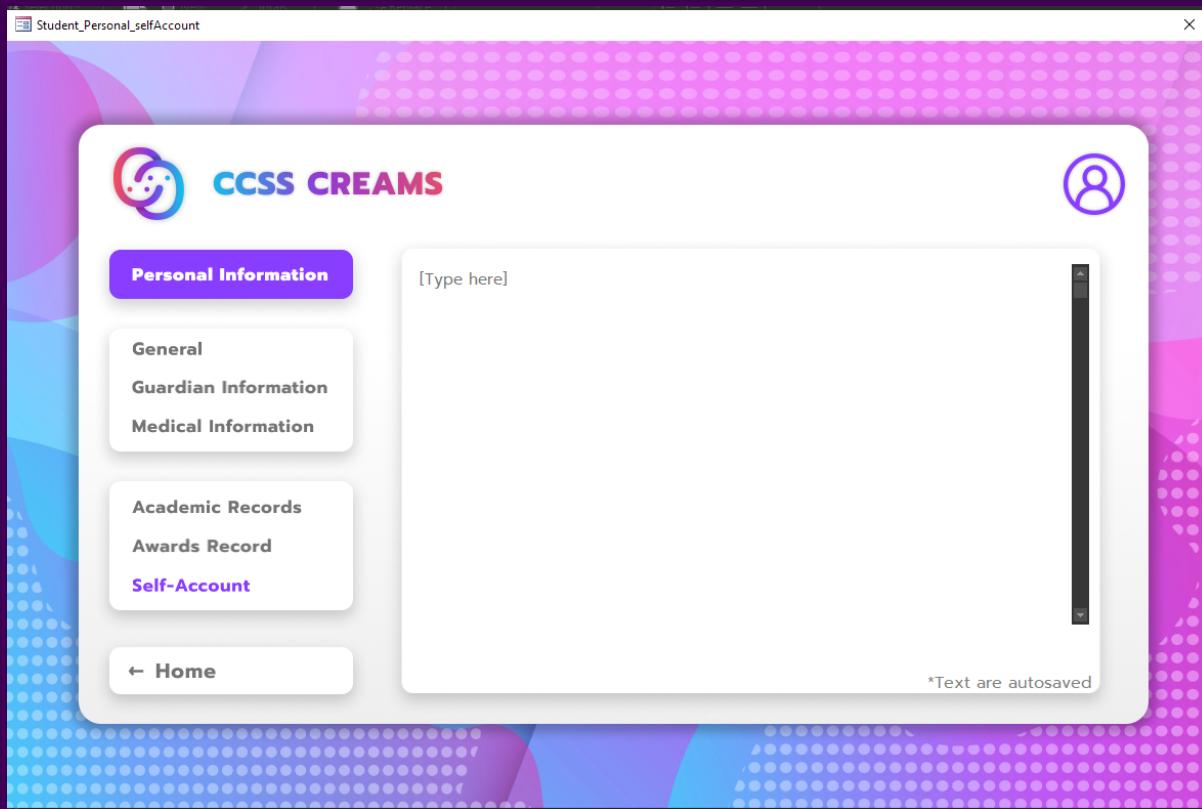
Private Sub btn_MedicalInformationOverlay_Click()
    ToPage "Student_Personal_medicalInformation"
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

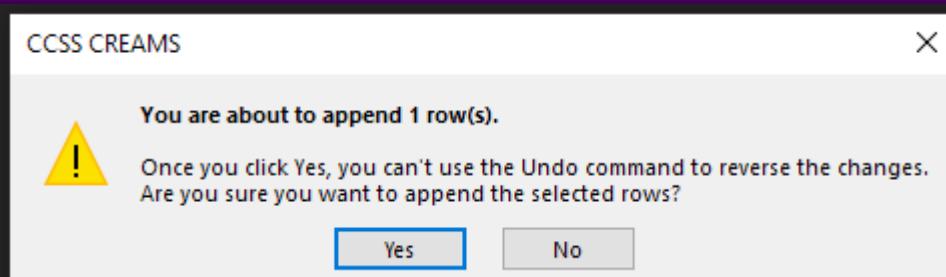
Private Sub btn_SelfAccountOverlay_Click()
    ToPage "Student_Personal_selfAccount"
End Sub

Private Sub Form_Load()
    Me.txt_stdid.Caption = getCurrentid
    DoCmd.Requery
End Sub
```

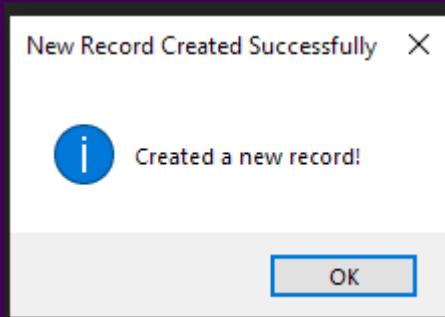
The last page of student personal is the student's self account, this is a form for students to type in their **self account**, all text typed will be **auto saved** in the database.



If a student has no save, the system will auto generate a new record for he/ she.



If a new record is added successfully, this message will pop up.



VBA of Student_Personal_selfAccount

```
Option Compare Database
Option Explicit

Private Sub btn_AcademicRecordsOverlay_Click()
    ToPage "Student_Personal_academicRecords"
End Sub

Private Sub btn_AwardsRecordOverlay_Click()
    ToPage "Student_Personal_awardsRecord"
End Sub

Private Sub btn_GeneralOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

Private Sub btn_GuardianInformationOverlay_Click()
    ToPage "Student_Personal_guardianInformation"
End Sub

Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

Private Sub btn_MedicalInformationOverlay_Click()
    ToPage "Student_Personal_medicalInformation"
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Student_Personal_general"
End Sub

Private Sub btn_SelfAccountOverlay_Click()
    ToPage "Student_Personal_selfAccount"
End Sub

Function selfAccountRecordCheck() As Boolean
    If Nz(DCount("[Student_selfAccount]", "Student_selfAccount",
    "[Student_id] = '" & [getCurrentid] & "'"), 0) = 0 Then
```

```
    selfAccountRecordCheck = False
    DoCmd.RunSQL "INSERT INTO Student_selfAccount (Student_id,
Student_selfAccount) VALUES ('" & getCurrentid & "', ' [Type
here] ' )"
    SuccessMessage (1)

    Else
        selfAccountRecordCheck = True

    End If

    Me.Refresh

End Function

Private Sub Form_Load()
    Me.txt_stdid.Caption = getCurrentid

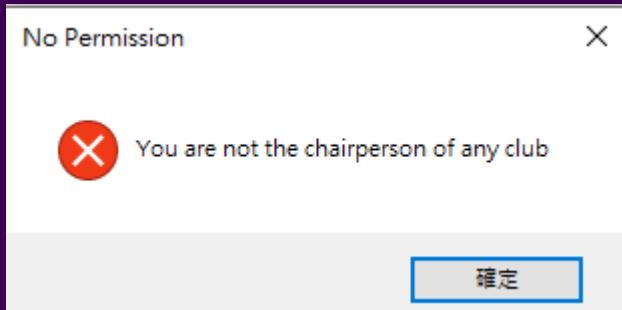
    If getCurrentid <> "" Then
        selfAccountRecordCheck

    Else
        ErrorMessage (3)

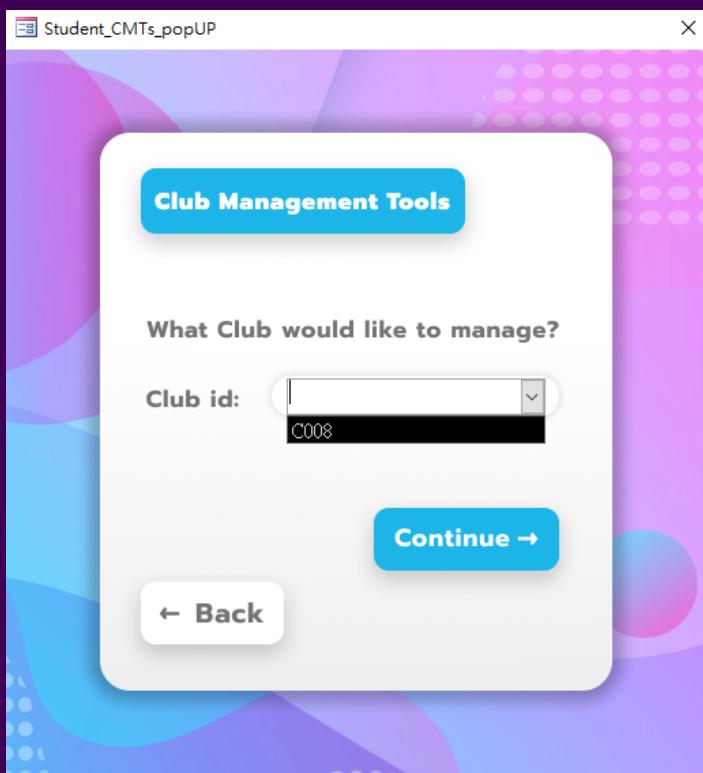
    End If

End Sub
```

The Last one is CMTs, CMTs stand for **club management tools**. It will open only if you are a chairman of a club. If you are **not a chairman** for a club it will appear as an **error message**.



If they are club **chairman** they will go to a Student_CMTs_popUp form and choose the Club_id that they want to manage.



VBA of Student_CMTs_popUp

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_ContinueOverlay_Click()
    setCurrentTMPid (Nz(ID, 0))
    If getCurrentTMPid = "0" Then
```

```
ErrorMessage (6)

Else
    ToTempPage "Student_CMTs_attendance"

End If

End Sub

Private Sub btn_HomeOverlay_Click()

    BackToStudentHome

End Sub


Private Sub Form_Timer()

    Me.TimerInterval = 0
    DoCmd.Close
    BackToStudentHome

End Sub


Private Sub Form_Load()

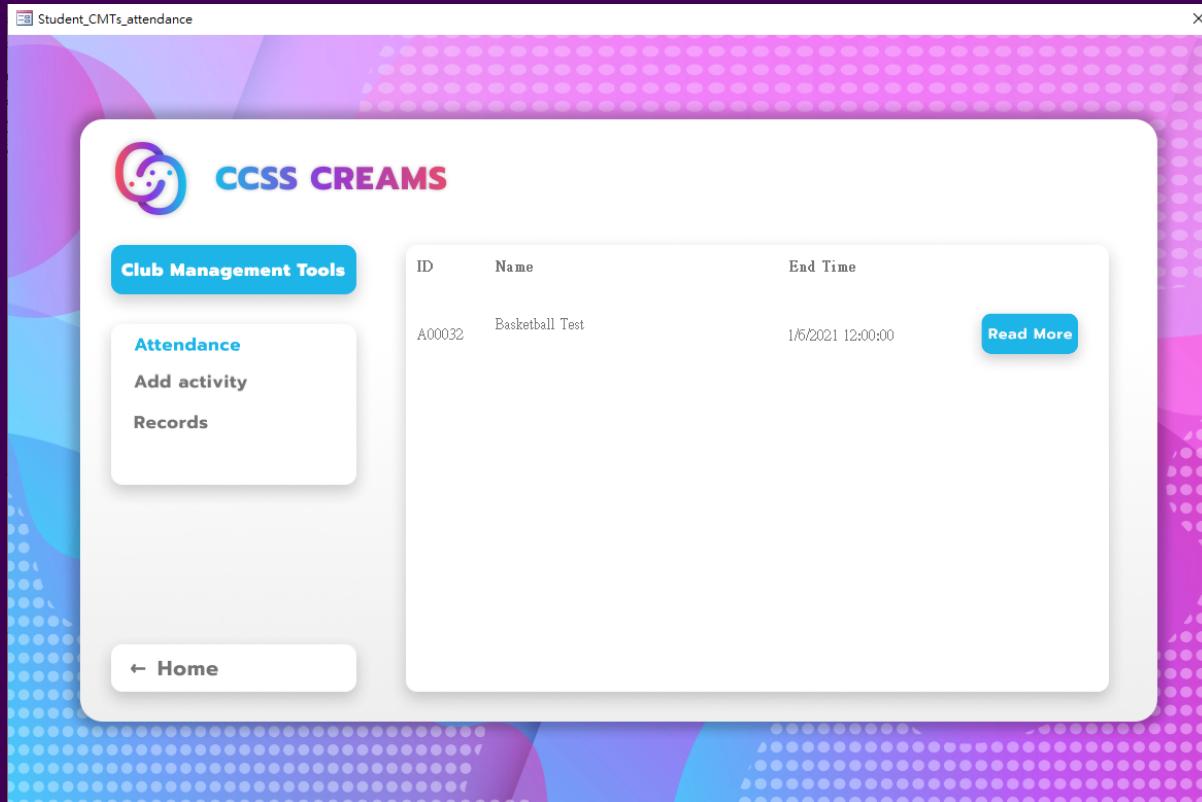
    If ((Nz((DCount("*", "Club_schoolyear", "Student_id = '" &
    [getCurrentid] & "'AND SchoolYear = '" & [getCurrentSchoolYear] &
    "'")), 0)) = 0) Then
        ErrorMessage (5)
        Me.TimerInterval = 50

    Else
        Me.TimerInterval = 0
        Me.txt_stdid.Caption = getCurrentid
        Me.txt_schoolyear.Caption = getCurrentSchoolYear

    End If

End Sub
```

When they press the continue button they will go to a form that list the **activity the club held** like this:



VBA of Student_CMTs_attendance

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_attendanceOverlay_Click()
```

```
    Me.Refresh
    DoCmd.Requery
```

```
End Sub
```

```
Private Sub btn_addActOverlay_Click()
```

```
    ToTempPage "Student_CMTs_addActivity"
```

```
End Sub
```

```
Private Sub btn_recordsOverlay_Click()
```

```
    ToTempPage "Student_CMTs_records"
```

```
End Sub
```

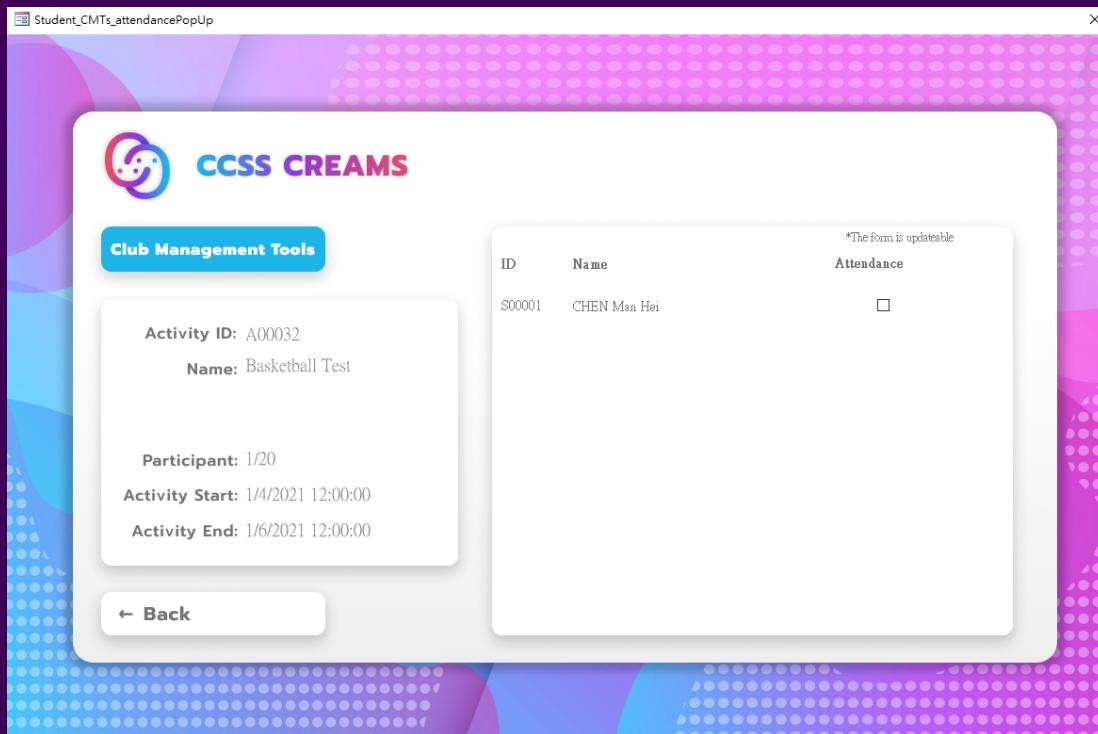
```
Private Sub btn_HomeLogoOverlay_Click()
    BackToStudentHome
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToStudentHome
End Sub
Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime

    Me.Refresh
    DoCmd.Requery
End Sub
```

After that they can click read more to go to Student_CMTs_attendancePopUp form and it will list the **participants of the student** and they can use it to take attendance, in order to **take attendance** they can **tick the box** next to the student name.



VBA of Student_CMTs_attendancePopUp

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_attendanceOverlay_Click()
    ToTempPage "Student_CMTs_attendance"
End Sub
```

```
Private Sub btn_addActOverlay_Click()
    ToTempPage "Student_CMTs_addActivity"
End Sub
```

```
Private Sub btn_recordsOverlay_Click()
    ToTempPage "Student_CMTs_records"
End Sub
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToStudentHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Student_CMTs_attendance"
End Sub

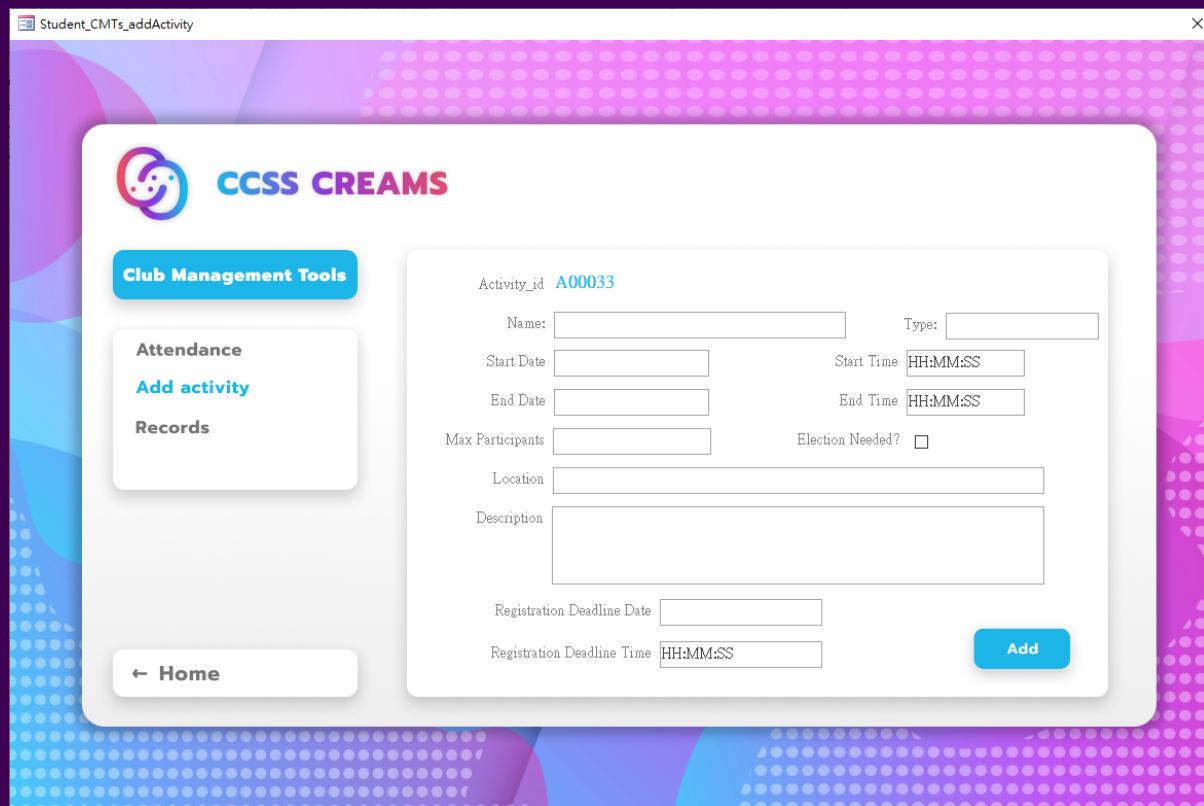
Private Sub btn_HomeTextOverlay_Click()
    BackToStudentHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime

    Me.act_id.Caption = getCurrentTMPid2
    Me.act_name.Caption = Nz(DLookup("[Activity_name]",
    "Activity_information", "[Activity_id] = '" & [getCurrentTMPid2]
    & "'", 0))
    Me.act_part.Caption = Nz(DCount("Activity_id",
    "Activity_record", "Activity_id = '" & getCurrentTMPid2 & "'") &
    "/" & Nz(DLookup("[Activity_maxParticipants]",
    "Activity_information", "[Activity_id] = '" & [getCurrentTMPid2]
    & "'", 0))
    Me.act_start.Caption = Nz(DLookup("[Activity_startDate]",
    "Activity_information", "[Activity_id] = '" & [getCurrentTMPid2]
    & "'", 0) & " " & Nz(DLookup("[Activity_startTime]",
    "Activity_information", "[Activity_id] = '" & [getCurrentTMPid2]
    & "'", 0))
    Me.act_end.Caption = Nz(DLookup("[Activity_endDate]",
    "Activity_information", "[Activity_id] = '" & [getCurrentTMPid2]
    & "'", 0) & " " & Nz(DLookup("[Activity_endTime]",
    "Activity_information", "[Activity_id] = '" & [getCurrentTMPid2]
    & "'", 0))

    Me.Refresh
    DoCmd.Requery
End Sub
```

The next function is Add activity and it will go to Student_CMTs_addActivity form and it is basically a form which is for **adding an activity representing the student's club.**



VBA of Student_CMTs_addActivity

```

Option Compare Database
Option Explicit

Private Sub btn_addOverlay_Click()

If IsNull(Me.act_id.Value) Or IsNull(Me.act_name.Value) Or
IsNull(Me.act_type.Value) Or IsNull(Me.act_startdate.Value) Or
IsNull(Me.act_starttime.Value) Or IsNull(Me.act_enddate.Value) Or
IsNull(Me.act_endtime.Value) Or IsNull(Me.act_loc.Value) Or
IsNull(Me.act_loc.Value) Or IsNull(Me.act_dec.Value) Or
IsNull(Me.act_regdeaddate.Value) Or
IsNull(Me.act_regdeadtime.Value) Or IsNull(Me.act_maxpart) Then
    ErrorMessage 8

ElseIf Nz(DCount("*", "Activity_information", "Activity_id='"
& Me.act_id.Value & "' AND club_id = '" & Me.txt_clubid.Caption &
"'"), 0) Then
    ErrorMessage 9

Else
    If MsgBox("Add activity?", vbYesNo + vbQuestion) = vbYes Then
        DoCmd.OpenQuery
        "Student_InsertRecords_Activity_information"
    End If
End If

```

```
If Nz(DCount("*", "Activity_information", "Activity_id='"
& Me.act_id.Value & "' AND club_id = '" & Me.txt_clubid.Caption &
"'"), 0) Then
    SuccessMessage 3

Else
    ErrorMessage 9

End If

Else
    MsgBox "Activity adding request canceled."

End If

End If

End Sub

Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.txt_teacher.Caption = Nz(DLookup("Teacher_id",
    "Club_schoolyear", "Club_id='"
    & getCurrentTMPid & "' AND
    SchoolYear='"
    & getCurrentSchoolYear & "'"), 0)

    Me.act_startdate.Value = ""
    Me.act_enddate.Value = ""
    Me.act_regdeaddate.Value = ""

    Me.act_starttime.Value = "HH:MM:SS"
    Me.act_endtime.Value = "HH:MM:SS"
    Me.act_regdeadtime = "HH:MM:SS"

    Dim Addid As Integer
    Dim LAddid As String
    Addid = (CInt(Right(DMax("Activity_id",
    "Activity_information"), 5)) + 1)
    LAddid = Len(CStr(Addid))

    If (LAddid = 1) Then
```

```
Me.act_id.Value = "A0000" & Addid

ElseIf (LAddid = 2) Then
    Me.act_id.Value = "A000" & Addid

ElseIf (LAddid = 3) Then
    Me.act_id.Value = "A00" & Addid

ElseIf (LAddid = 4) Then
    Me.act_id.Value = "A0" & Addid

ElseIf (LAddid = 5) Then
    Me.act_id.Value = "A" & Addid

Else
    ErrorMessage (99)

End If

Me.Refresh
'DoCmd.Requery

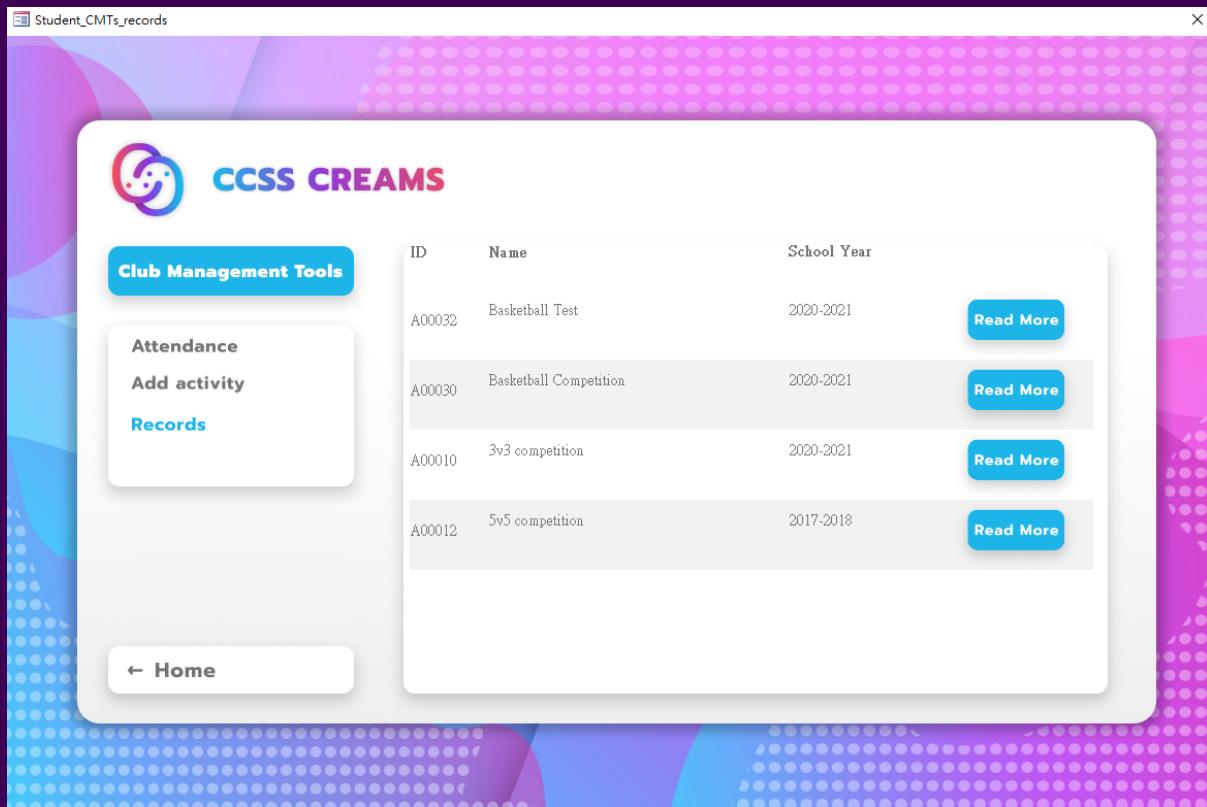
End Sub

Private Sub btn_attendanceOverlay_Click()
    ToTempPage "Student_CMTs_attendance"
End Sub

Private Sub btn_addActOverlay_Click()
    ToTempPage "Student_CMTs_addActivity"
End Sub

Private Sub btn_recordsOverlay_Click()
    ToTempPage "Student_CMTs_records"
End Sub
```

The Next one is The Records button and it will bring the student to the Student_CMTs_records, this form is to **list all the activity the club had held before**.



VBA of Student_CMTs_records

```
Option Compare Database
Option Explicit
```

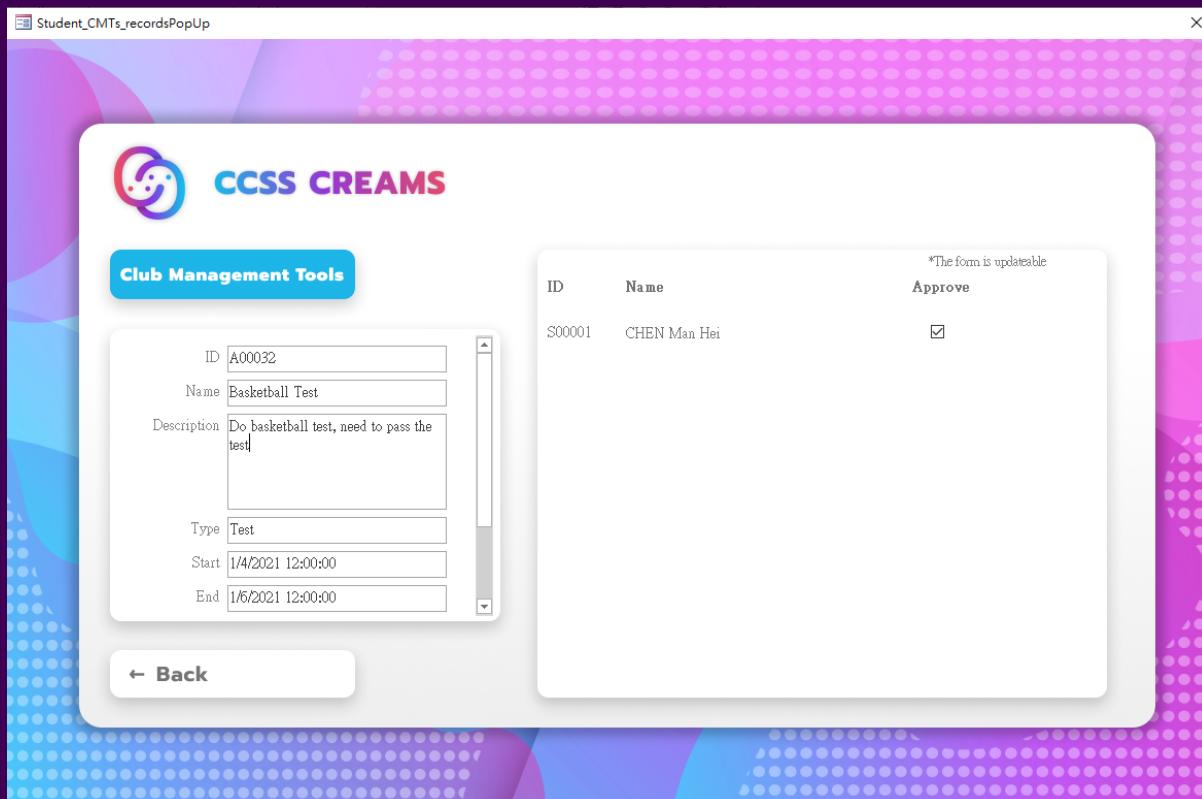
```
Private Sub btn_HomeOverlay_Click()
    BackToStudentHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.Refresh
    DoCmd.Requery
End Sub
```

```
Private Sub btn_attendanceOverlay_Click()
```

```
ToTempPage "Student_CMTs_attendance"  
End Sub  
  
Private Sub btn_addActOverlay_Click()  
    ToTempPage "Student_CMTs_addActivity"  
End Sub  
Private Sub btn_HomeTextOverlay_Click()  
    BackToStudentHome  
End Sub  
  
Private Sub btn_recordsOverlay_Click()  
    Me.Refresh  
    DoCmd.Requery  
End Sub
```

When the student presses Read More of one of the activities, it will bring the student to a new form called Student_CMTs_recordsPopup this is for the student chairman to show things of the activities. On the **left** hand side it shows all the **activity information** and on the **right** side it shows all the **student attendance**.



VBA of Student_CMTs_recordsPopup

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_attendanceOverlay_Click()
    ToTempPage "Student_CMTs_attendance"
End Sub

Private Sub btn_addActOverlay_Click()
    ToTempPage "Student_CMTs_addActivity"
End Sub

Private Sub btn_recordsOverlay_Click()
    ToTempPage "Student_CMTs_records"
End Sub
```

```
End Sub

Private Sub btn_HomeLogoOverlay_Click()
    BackToStudentHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Student_CMTs_records"
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToStudentHome
End Sub

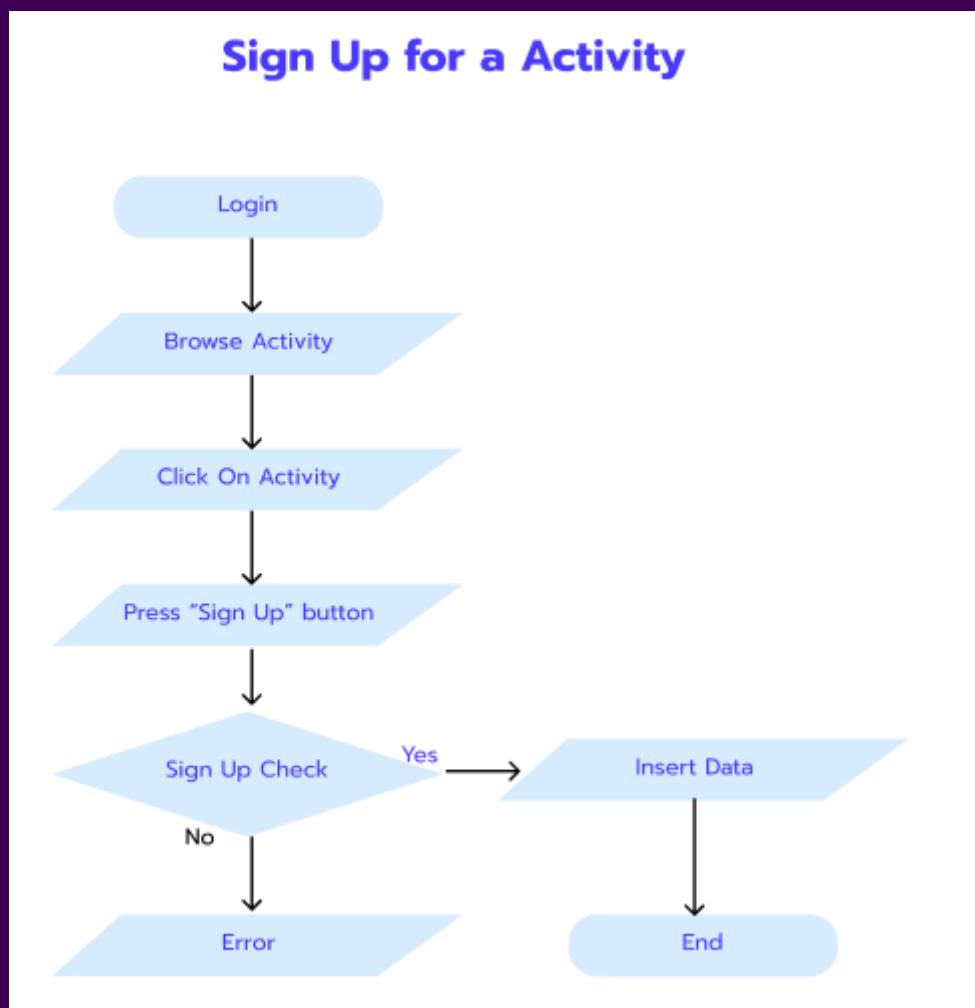
Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.txt_actid.Caption = getCurrentTMPid2

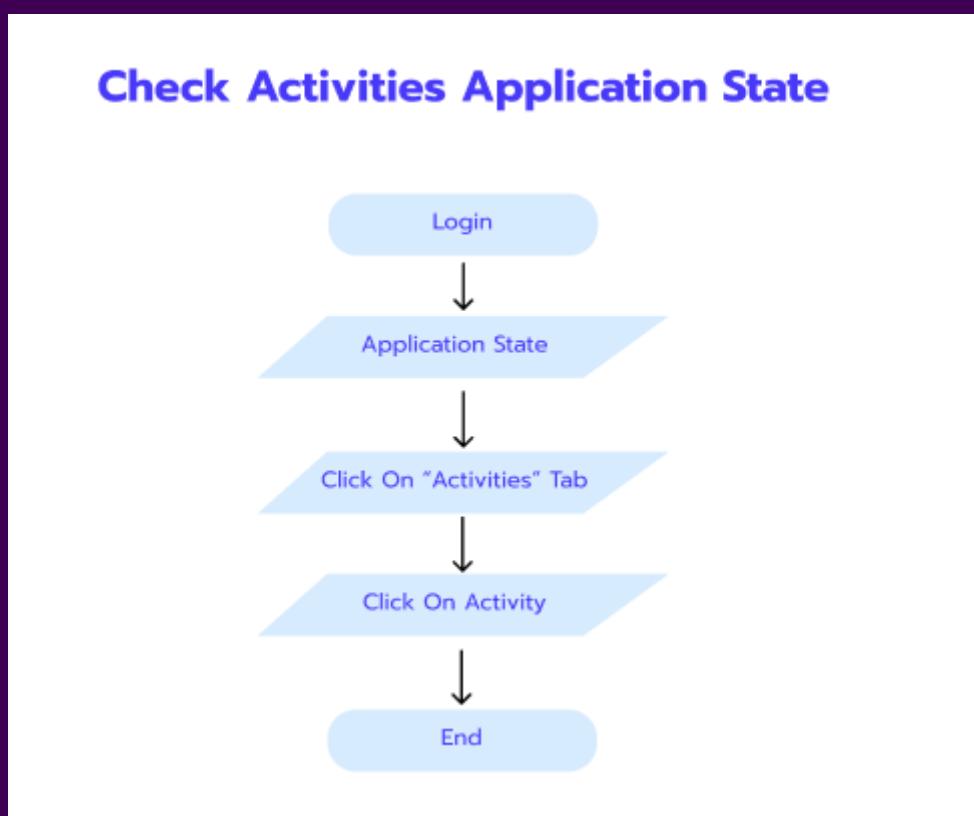
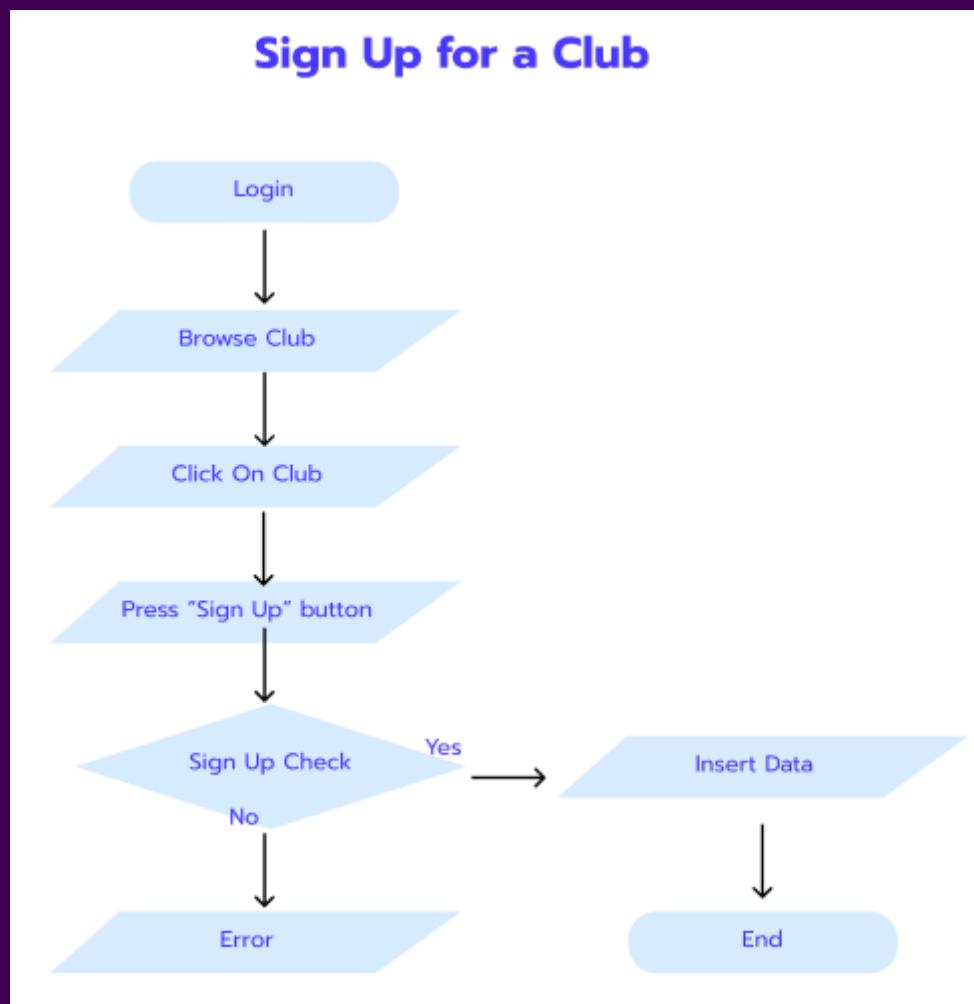
    Me.Refresh
    DoCmd.Requery
End Sub
```

They can press Back or the HKCC CREAMS icon to exit the CMTs.

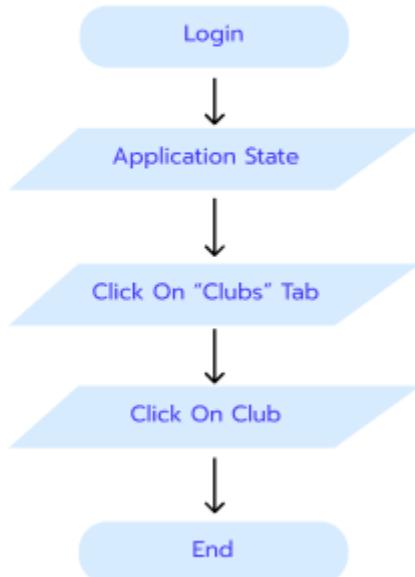
This the end for the student side of this activity system.

This means students can do a couple of functions. The flow of the functions of students:



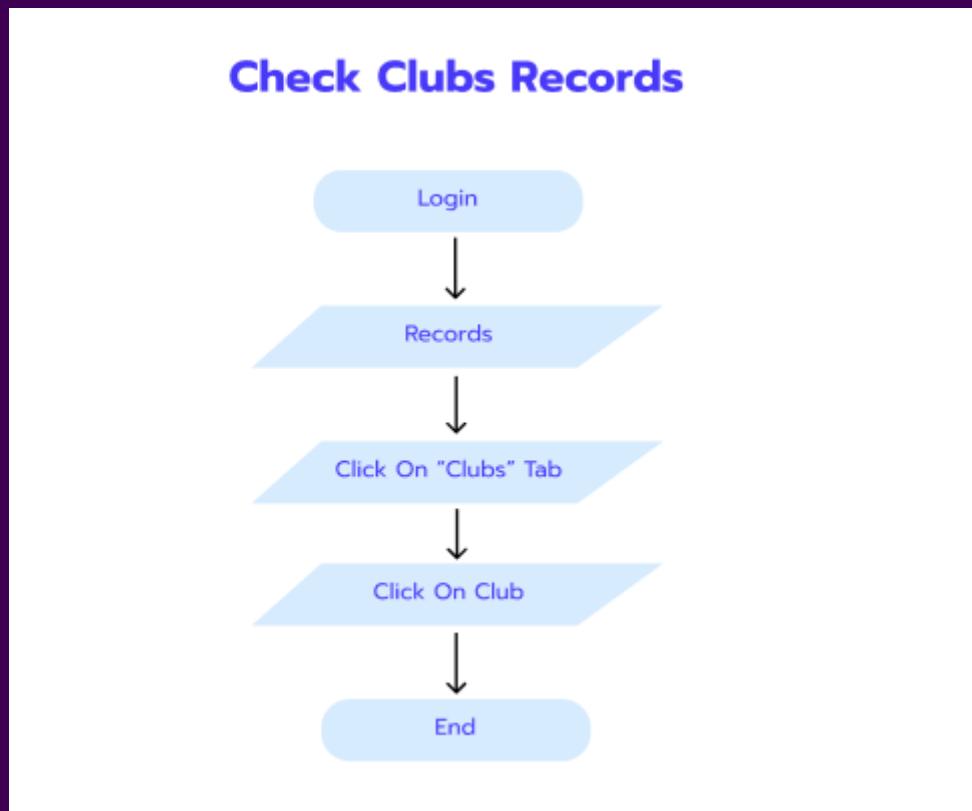


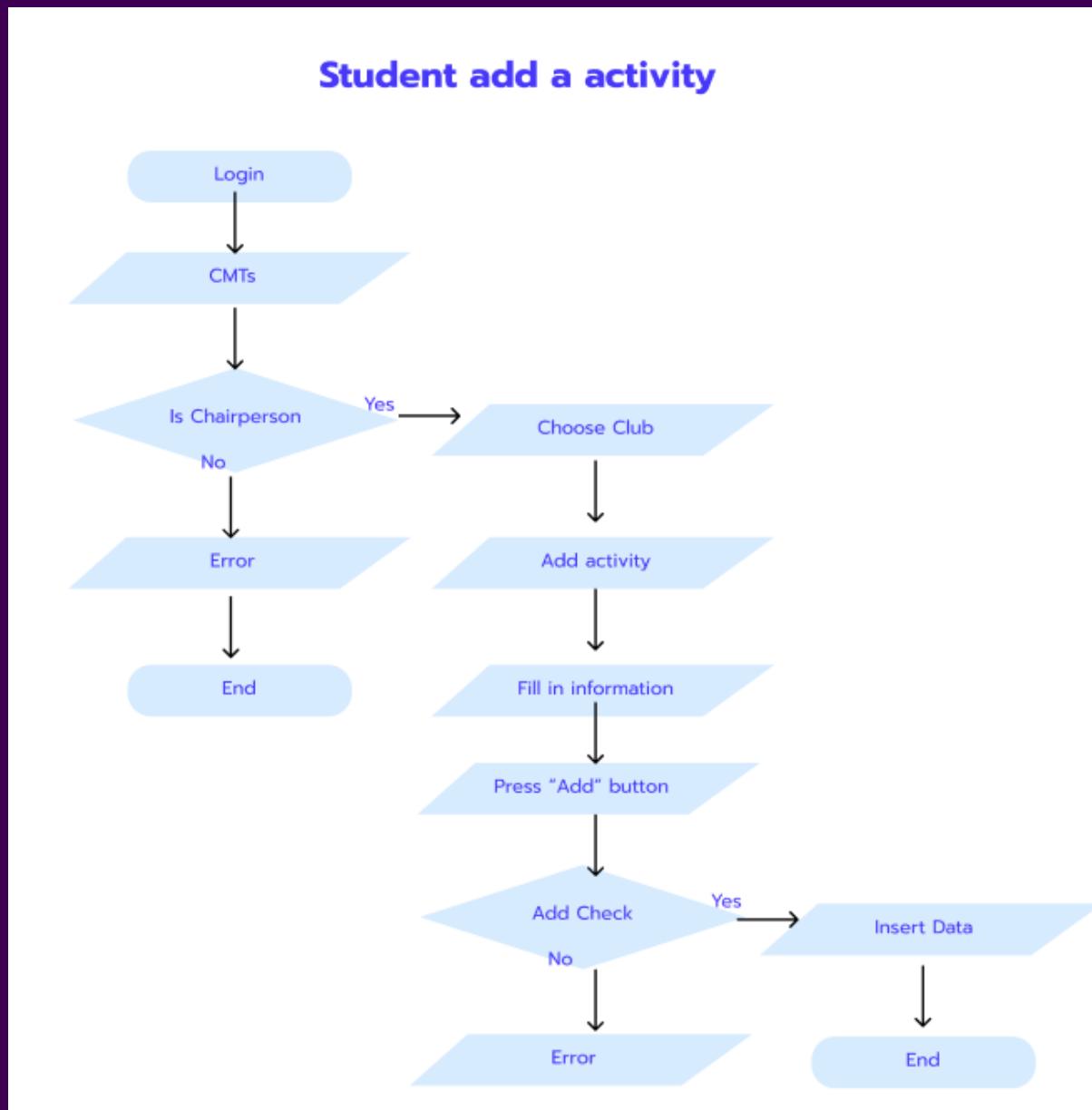
Check Clubs Application State



Check Activities Records

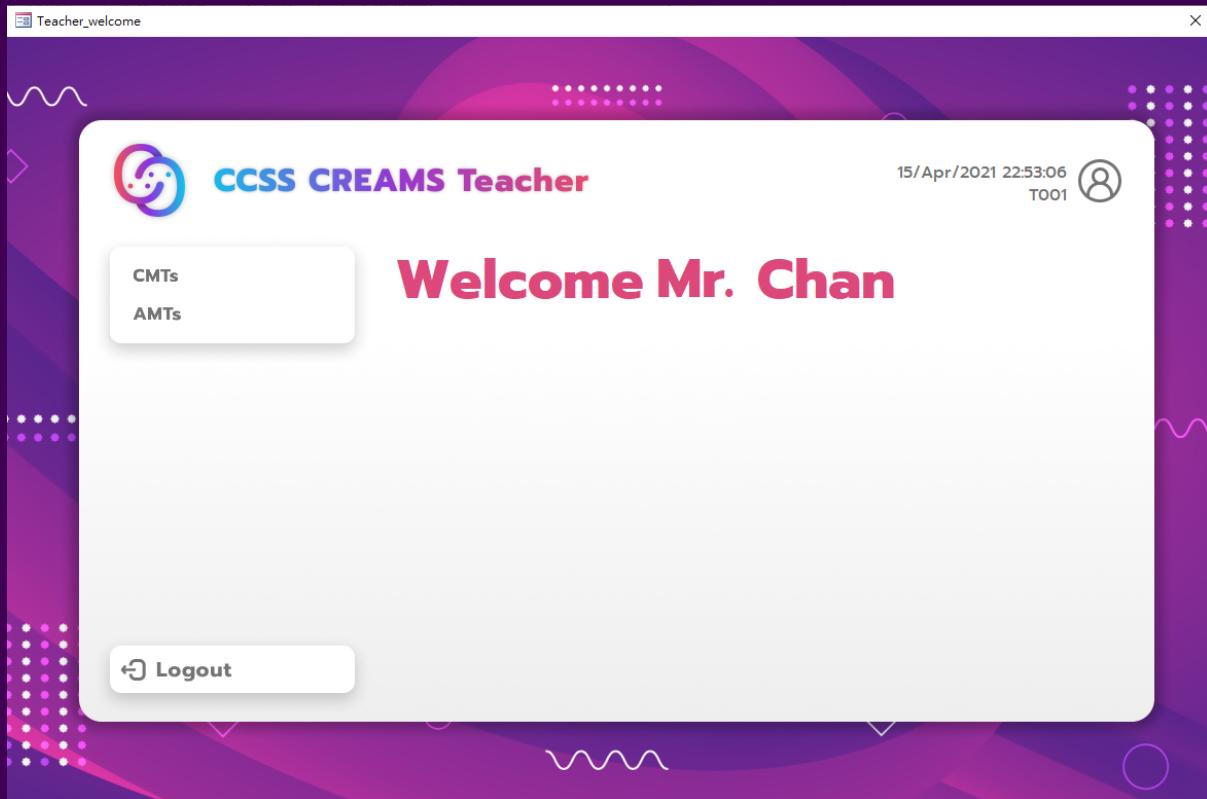




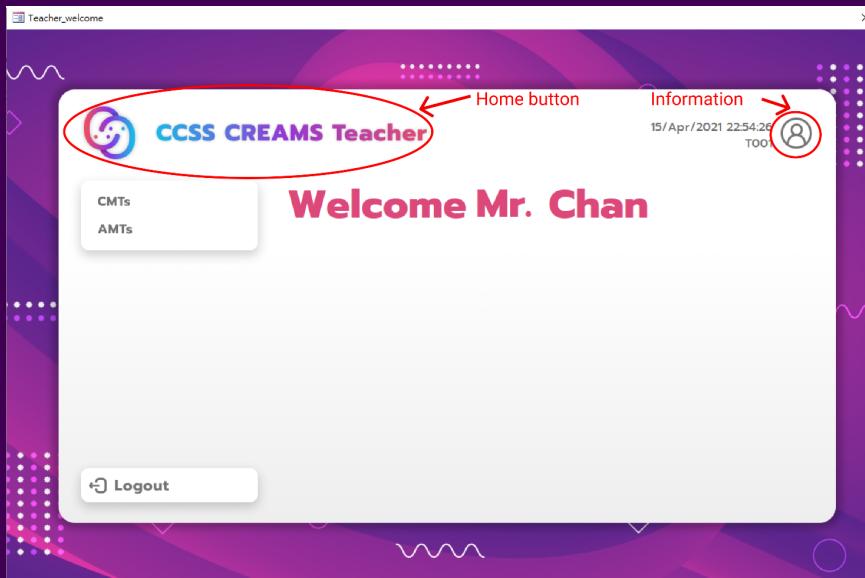


(c) Teacher side

When a teacher logs in to the system, it will bring the teacher to Teacher_welcome form. It will show a **different background** than the student.



Just like the student side the CCSS CREAMS Teacher logo can go to Teacher_welcome form and the top right circle can go to personal information of teacher

**VBA of Teacher_welcome**

Option Compare Database

```
Private Sub btn_AMTsOverlay_Click()
    ToTempPage "Teacher_AMTs_addActivity"
End Sub

Private Sub btn_CMTsOverlay_Click()
    ToTempPage "Teacher_CMTs_popUP"
End Sub

Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_LogoutOverlay_Click()
    ToPage "allLogin"
    setCurrentid ("")
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Teacher_Personal_general"
```

```
End Sub

Private Sub Form_Load()
    Dim CurrentTeacherGender As String

    Me.txt_tecid.Caption = getCurrentid
    Me.std_lname.Caption = Nz(DLookup("[Teacher_lastName]",
    "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'"), 0)
    CurrentTeacherGender = Nz(DLookup("[Teacher_gender]",
    "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'"), 0)

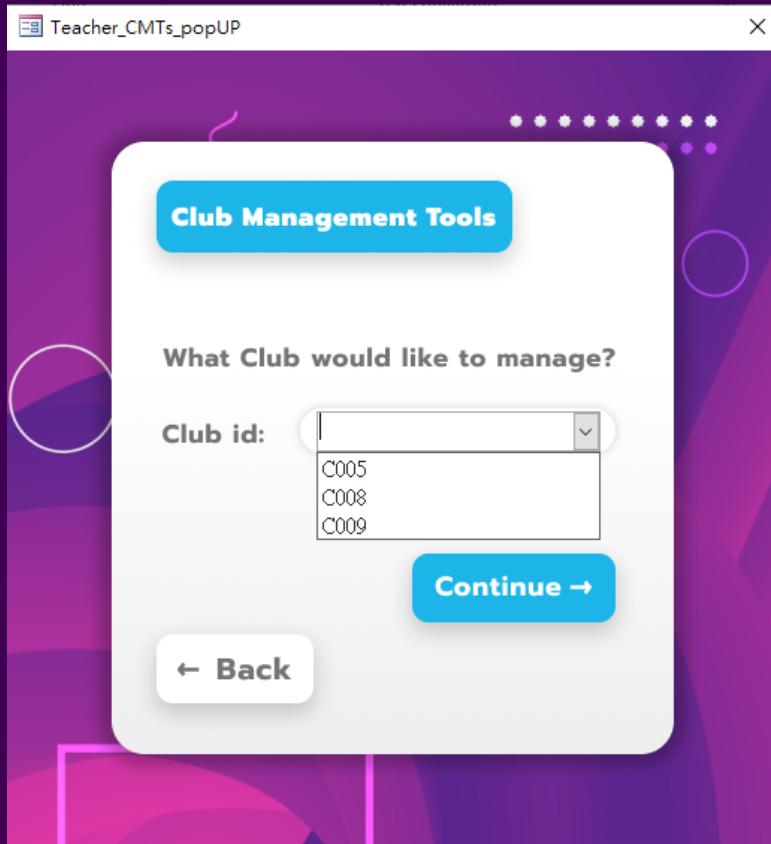
    If CurrentTeacherGender = "F" Then
        Me.std_gender.Caption = "Ms."
    Else
        Me.std_gender.Caption = "Mr."
    End If

    Me.txt_Clock.Caption = getNowDate & " " & getNowTime

End Sub

Private Sub Form_Timer()
    Me.txt_Clock.Caption = getNowDate & " " & getNowTime
End Sub
```

When going to CMTs, it will go to the Teacher_CMTspopUP form and just like the student side it will **show all the clubs that the teacher manages**.



VBA of Teacher_CMTspopUP

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_ContinueOverlay_Click()
    setCurrentTMPid (Nz(ID, 0))

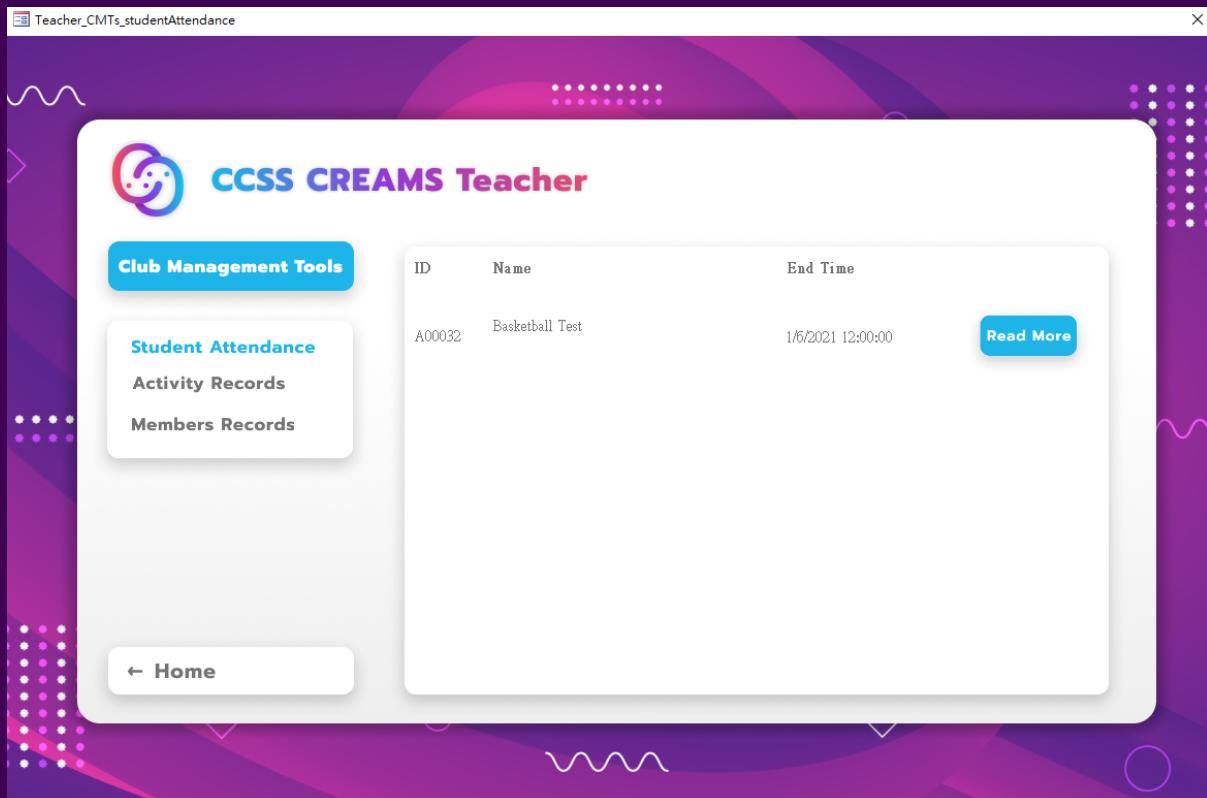
    If getCurrentTMPid = "0" Then
        ErrorMessage (6)

    Else
        ToTempPage "Teacher_CMTs_studentAttendance"
    End If
End Sub

Private Sub btn_HomeOverlay_Click()
    BackToTeacherHome
End Sub
```

```
Private Sub Form_Timer()  
  
    Me.TimerInterval = 0  
    DoCmd.Close  
    BackToStudentHome  
  
End Sub  
  
  
Private Sub Form_Load()  
  
    If ((Nz((DCount("*", "Club_schoolyear", "Teacher_id = '" &  
[getCurrentid] & "'AND SchoolYear = '" & [getCurrentSchoolYear] & "'')),  
0)) = 0) Then  
        ErrorMessage (7)  
        Me.TimerInterval = 50  
  
    Else  
        Me.TimerInterval = 0  
        Me.txt_teaid.Caption = getCurrentid  
        Me.txt_schoolyear.Caption = getCurrentSchoolYear  
  
    End If  
  
End Sub
```

After pressing continue, it will go to Teacher_CMTs_studentAttendance form and it will **list activities that were held**.



VBA of Teacher_CMTs_studentAttendance

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_actRecOverlay_Click()
    ToTempPage "Teacher_CMTs_activityRecords"
End Sub

Private Sub btn_attOverlay_Click()
    ToTempPage "Teacher_CMTs_studentAttendance"
End Sub

Private Sub btn_recordsOverlay_Click()
    ToTempPage "Teacher_CMTs_memberRecords"
End Sub

Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub
```

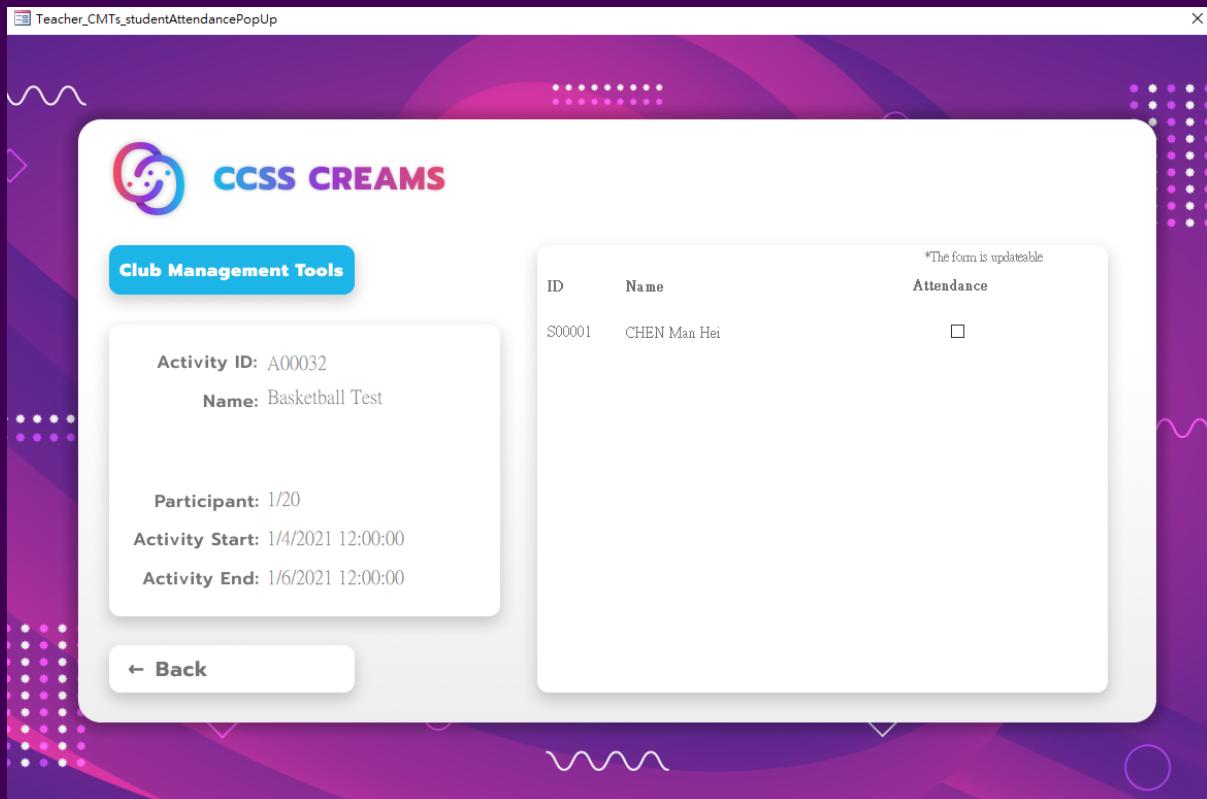
```
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub
Private Sub btn_HomeOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime

    Me.Refresh
    DoCmd.Requery
End Sub
```

Is like the student side, when the teacher presses more and the Teacher_CMTs_studentAttendancePopUp will pop up and it will list the participants and the teacher can **tick the box** next to the student name in order to **take attendance**.



VBA of Teacher_CMTs_studentAttendancePopUp

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Teacher_CMTs_studentAttendance"
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub
```

```
Private Sub Form_Load()

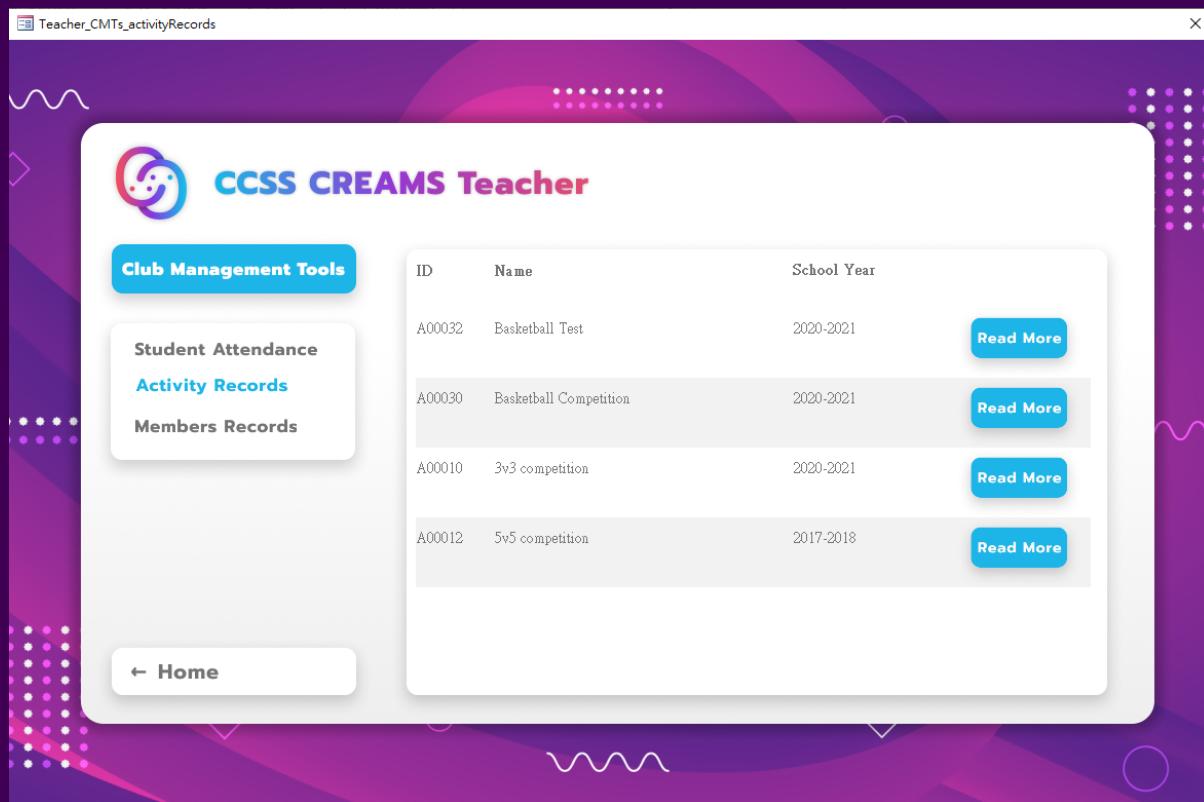
    Me.txt_clubid.Caption = getCurrentTPMpid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime

    Me.act_id.Caption = getCurrentTPMpid2
    Me.act_name.Caption = Nz(DLookup("[Activity_name]", "Activity_information", "[Activity_id] = '" & [getCurrentTPMpid2] & "'"), 0)
    Me.act_part.Caption = Nz(DCount("Activity_id", "Activity_record", "Activity_id = '" & getCurrentTPMpid2 & "')") & "/" &
    Nz(DLookup("[Activity_maxParticipants]", "Activity_information", "[Activity_id] = '" & [getCurrentTPMpid2] & "'"), 0)
    Me.act_start.Caption = Nz(DLookup("[Activity_startDate]", "Activity_information", "[Activity_id] = '" & [getCurrentTPMpid2] & "'"), 0) & " " & Nz(DLookup("[Activity_startTime]", "Activity_information", "[Activity_id] = '" & [getCurrentTPMpid2] & "'"), 0)
    Me.act_end.Caption = Nz(DLookup("[Activity_endDate]", "Activity_information", "[Activity_id] = '" & [getCurrentTPMpid2] & "'"), 0) & " " & Nz(DLookup("[Activity_endTime]", "Activity_information", "[Activity_id] = '" & [getCurrentTPMpid2] & "'"), 0)

    Me.Refresh
    DoCmd.Requery

End Sub
```

The next one is Activity Records, it is in the form which is called Teacher_CMTs_activityRecords which **lists all the different activities** that particular club held.



VBA of Teacher_CMTs_activityRecords

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

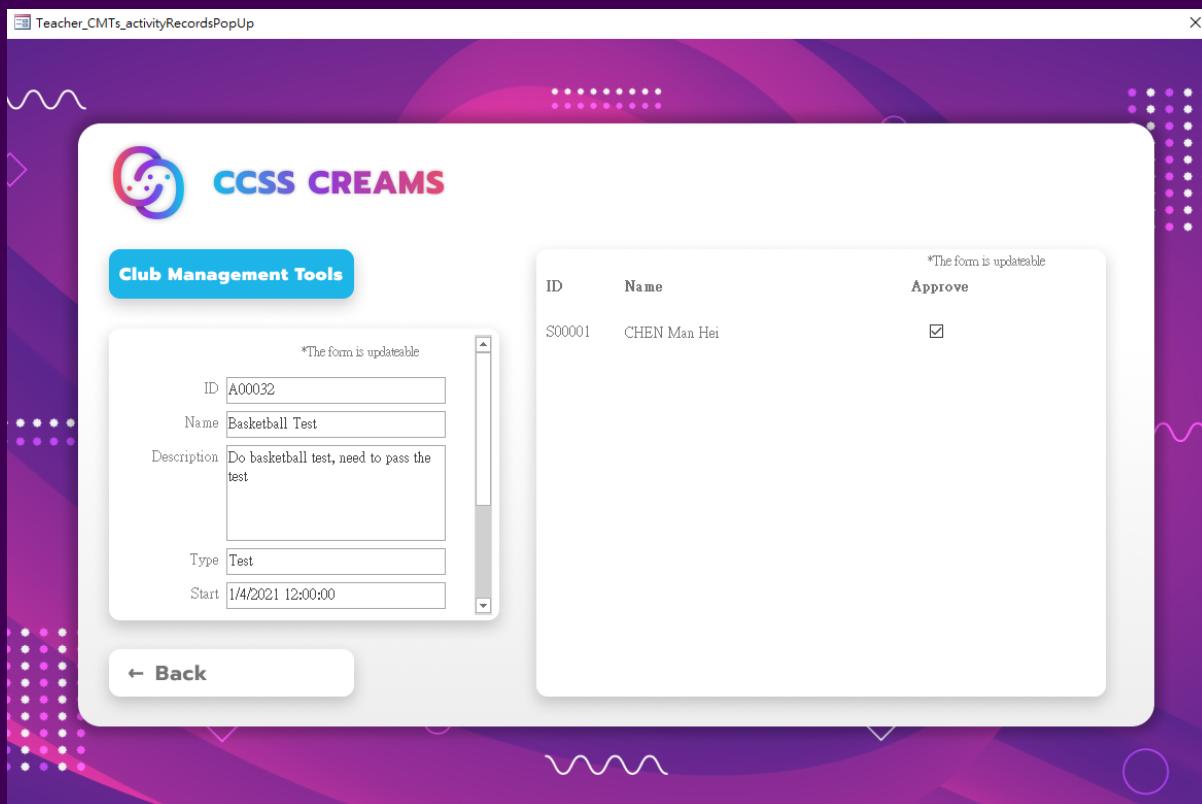
Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub
Private Sub btn_HomeOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_nav1Overlay_Click()
    ToTempPage "Teacher_AMTs_addActivity"

```


If the teacher presses the read more button, it brings the teacher to Teacher_CMTs_activityRecordsPopUp form and the teacher can modify the activities there.

The **left** hand side is the **activity information** and the **right** hand side is the **activity attendance**.



VBA of Teacher_CMTs_activityRecordsPopUp

```
Option Compare Database
Option Explicit
```

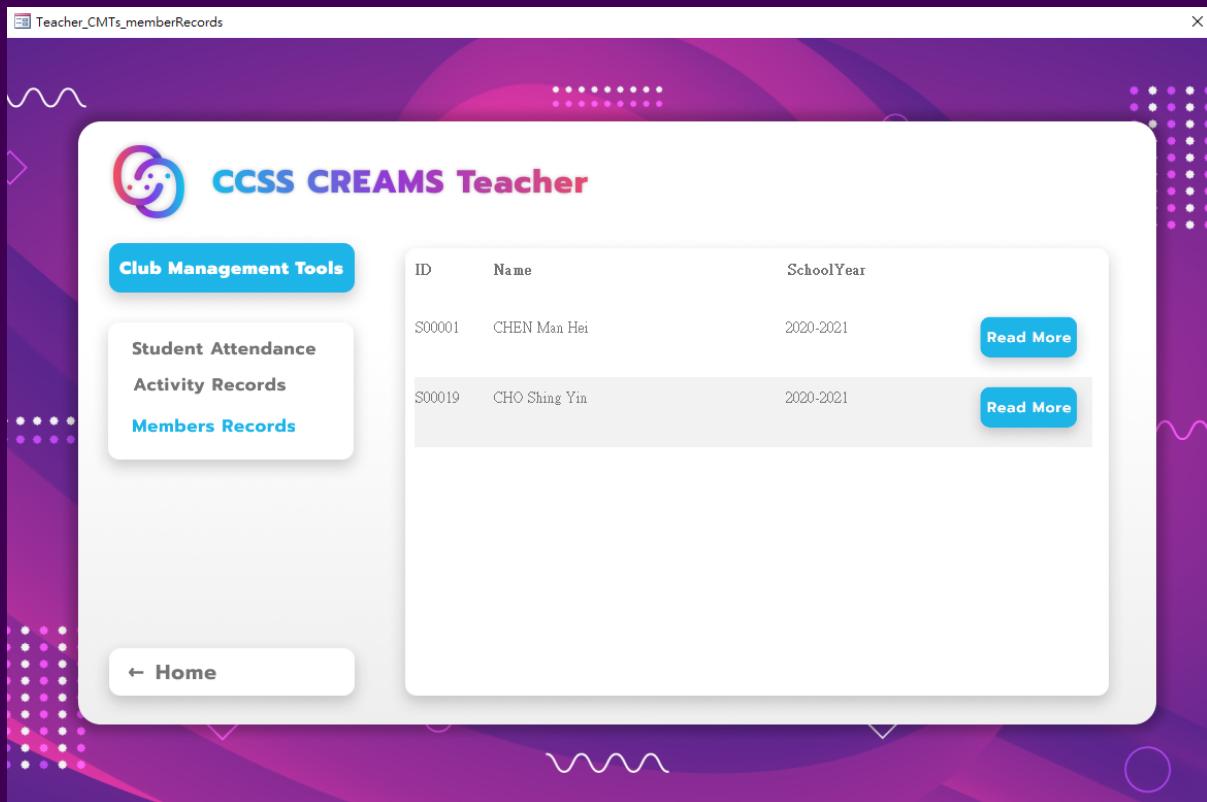
```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Teacher_CMTs_activityRecords"
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub
```

```
Private Sub Form_Load()  
  
    Me.txt_clubid.Caption = getCurrentTMPid  
    Me.txt_schoolyear.Caption = getCurrentSchoolYear  
    Me.txt_nowDate.Caption = getNowDate  
    Me.txt_nowTime.Caption = getNowTime  
    Me.txt_actid.Caption = getCurrentTMPid2  
  
    Me.Refresh  
    DoCmd.Requery  
  
End Sub
```

The next one is when the teacher **clicks** the **Members Records**, it will bring them to the Teacher_CMTs_memberRecords form and it will **show the club member list**.



VBA of Teacher_CMTs_activityRecordsPopUp

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Teacher_CMTs_activityRecords"
End Sub

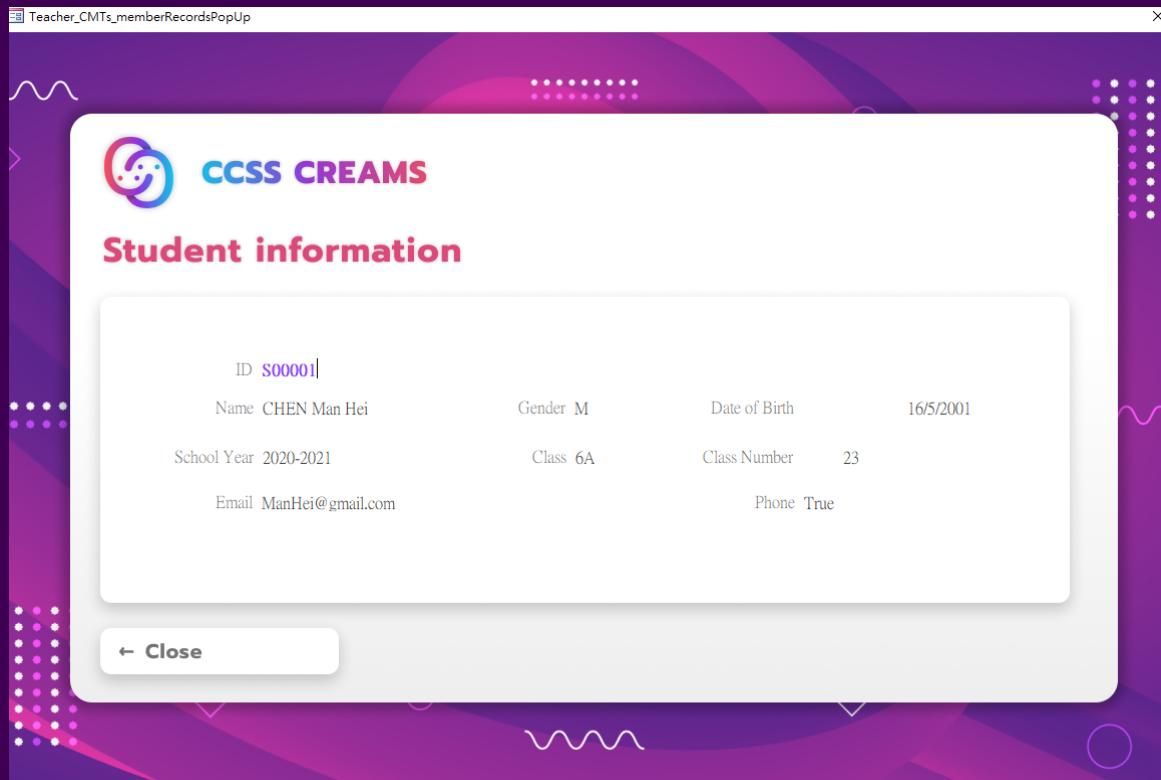
Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid

```

```
Me.txt_schoolyear.Caption = getCurrentSchoolYear  
Me.txt_nowDate.Caption = getNowDate  
Me.txt_nowTime.Caption = getNowTime  
Me.txt_actid.Caption = getCurrentTMPid2  
  
Me.Refresh  
DoCmd.Requery  
  
End Sub
```

Then the teacher can press **Read More** to the Teacher_CMTs_member see the further information of each member of the Student.



VBA of Teacher_CMTs_memberRecordsPopUp

```
Option Compare Database
Option Explicit
```

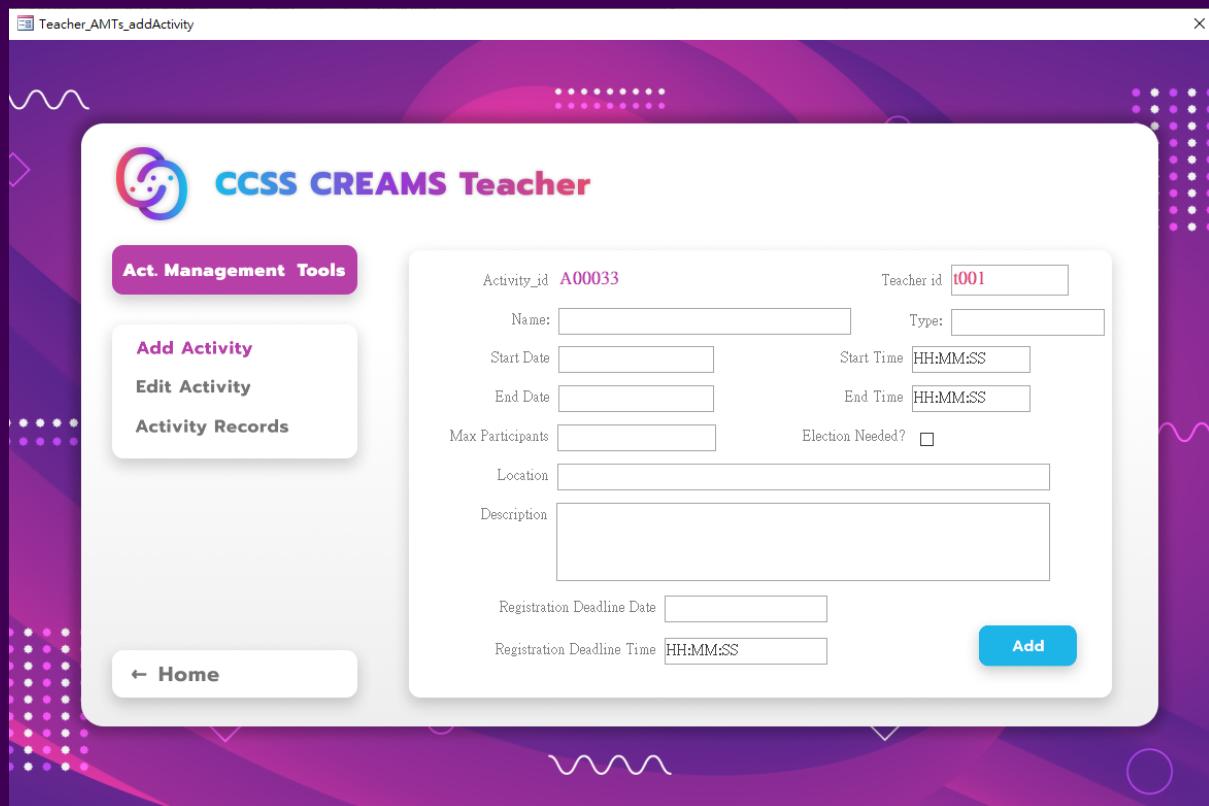
```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Teacher_CMTs_memberRecords"
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub Form_Load()
    Me.txt_schoolyear.Caption = CurrentTMPid3
    Me.txt_stdid.Caption = getCurrentTMPid2
    Me.Refresh
    DoCmd.Requery
End Sub
```

Then is the AMTs part, when a teacher clicks the AMTs, it will bring to the Teacher_AMTs_addActivity form and this will **create an activity** that does **not** have a **club in charge** of the activity. And after they press the Add button the activity will be added in the Activity_information record.



VBA of Teacher_AMTs_addActivity

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub
Private Sub btn_HomeOverlay_Click()
    BackToTeacherHome
End Sub
```

```
Private Sub btn_nav1Overlay_Click()
    ToTempPage "Teacher_AMTs_addActivity"
End Sub

Private Sub btn_nav2Overlay_Click()
    ToTempPage "Teacher_AMTs_editActivity"
End Sub

Private Sub btn_nav3Overlay_Click()
    ToTempPage "Teacher_AMTs_activityRecords"
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_schoolyear.Caption = getCurrentSchoolYear
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime
    Me.txt_teacher.Value = getCurrentid

    Me.act_startdate.Value = ""
    Me.act_enddate.Value = ""
    Me.act_regdeaddate.Value = ""

    Me.act_starttime.Value = "HH:MM:SS"
    Me.act_endtime.Value = "HH:MM:SS"
    Me.act_regdeadtime = "HH:MM:SS"

    Dim Addid As Integer
    Dim LAddid As String
    Addid = (CInt(Right(DMax("Activity_id", "Activity_information"), 5)) +
1)
    LAddid = Len(CStr(Addid))

    If (LAddid = 1) Then
        Me.act_id.Value = "A0000" & Addid

    ElseIf (LAddid = 2) Then
        Me.act_id.Value = "A000" & Addid

    ElseIf (LAddid = 3) Then
        Me.act_id.Value = "A00" & Addid

    ElseIf (LAddid = 4) Then
        Me.act_id.Value = "A0" & Addid

    ElseIf (LAddid = 5) Then
        Me.act_id.Value = "A" & Addid

    Else
        ErrorMessage (99)
    End If
```

```
Me.Refresh
'DoCmd.Requery

End Sub

Private Sub btn_addOverlay_Click()

    If IsNull(Me.act_id.Value) Or IsNull(Me.act_name.Value) Or
    IsNull(Me.act_type.Value) Or IsNull(Me.act_startdate.Value) Or
    IsNull(Me.act_starttime.Value) Or IsNull(Me.act_enddate.Value) Or
    IsNull(Me.act_endtime.Value) Or IsNull(Me.act_loc.Value) Or
    IsNull(Me.act_loc.Value) Or IsNull(Me.act_dec.Value) Or
    IsNull(Me.act_regdeaddate.Value) Or IsNull(Me.act_regdeadtime.Value) Or
    IsNull(Me.act_maxpart.Value) Or IsNull(Me.txt_teacher.Value) Then
        ErrorMessage 8

    ElseIf Nz(DCount("*", "Activity_information", "Activity_id='"
    & Me.act_id.Value & "' AND club_id = '" & Me.txt_clubid.Caption & "')"), 0)
    Then
        ErrorMessage 9

    Else
        If MsgBox("Add activity?", vbYesNo + vbQuestion) = vbYes Then
            DoCmd.OpenQuery "Teacher_InsertRecords_Activity_information"

                If Nz(DCount("*", "Activity_information", "Activity_id='"
    & Me.act_id.Value & "' AND club_id = '" & Me.txt_clubid.Caption & "')"), 0)
    Then
                    SuccessMessage 3

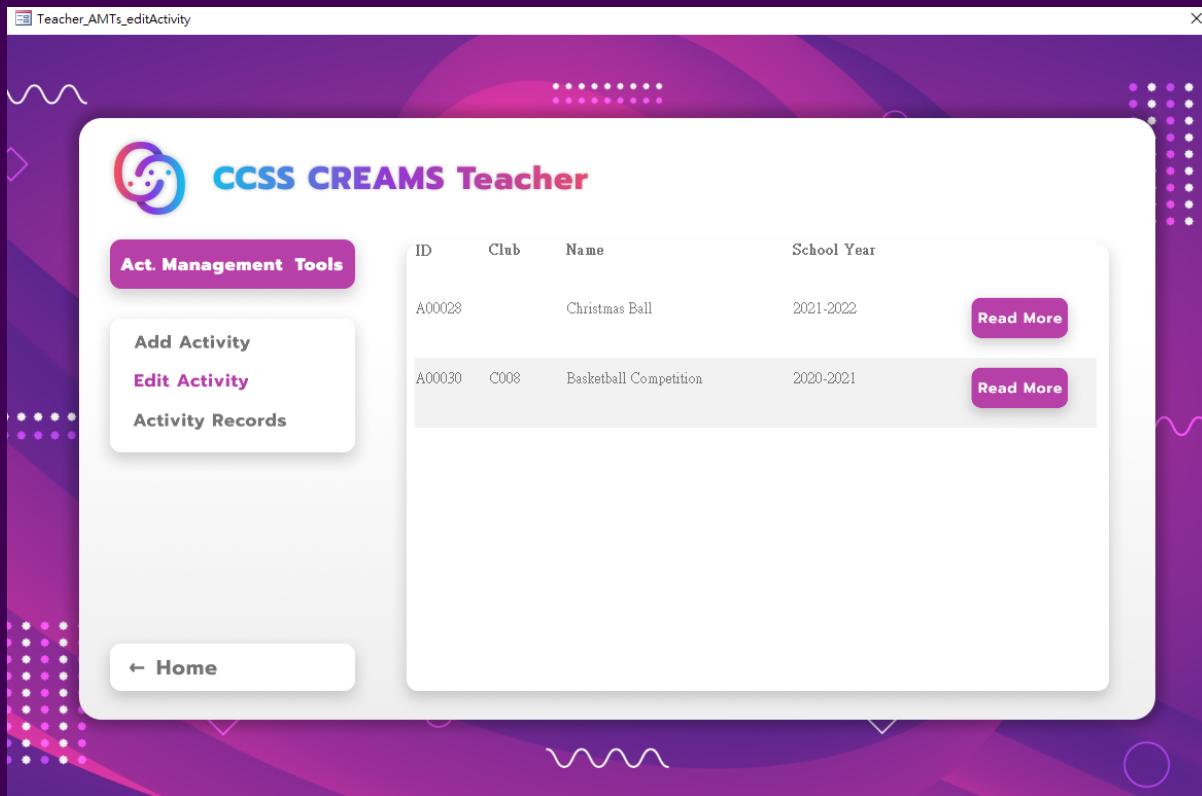
                Else
                    ErrorMessage 9

            End If

        Else
            MsgBox "Activity adding request canceled."
        End If

    End If
End Sub
```

The next function in the AMTs is Edit Activity and when they press the Edit Activity button it will show the activity that is in charge by that particular teacher.



VBA of Teacher_AMTs_editActivity

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub
Private Sub btn_HomeOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_nav1Overlay_Click()
    ToTempPage "Teacher_AMTs_addActivity"
End Sub
```

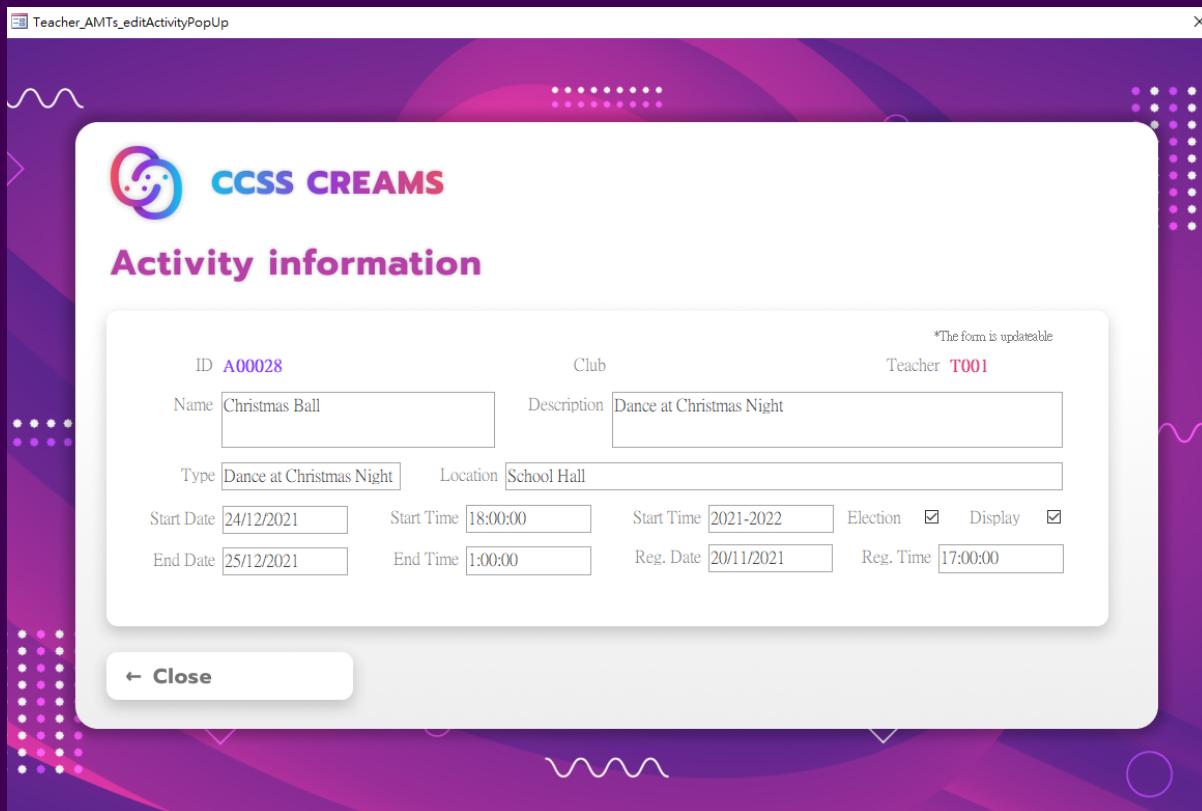
```
Private Sub btn_nav2Overlay_Click()
    ToTempPage "Teacher_AMTs_editActivity"
End Sub

Private Sub btn_nav3Overlay_Click()
    ToTempPage "Teacher_AMTs_activityRecords"
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid
    Me.txt_teacherid.Caption = getCurrentid
    Me.txt_nowDate.Caption = getNowDate
    Me.txt_nowTime.Caption = getNowTime

    Me.Refresh
    'DoCmd.Requery
End Sub
```

After that teacher can click **Read More** to modify activity, it will go to the Teacher_AMTs_editActivityPopUp form to **edit the selected activity**.



VBA of Teacher_AMTs_editActivityPopUp

```
Option Compare Database
Option Explicit
```

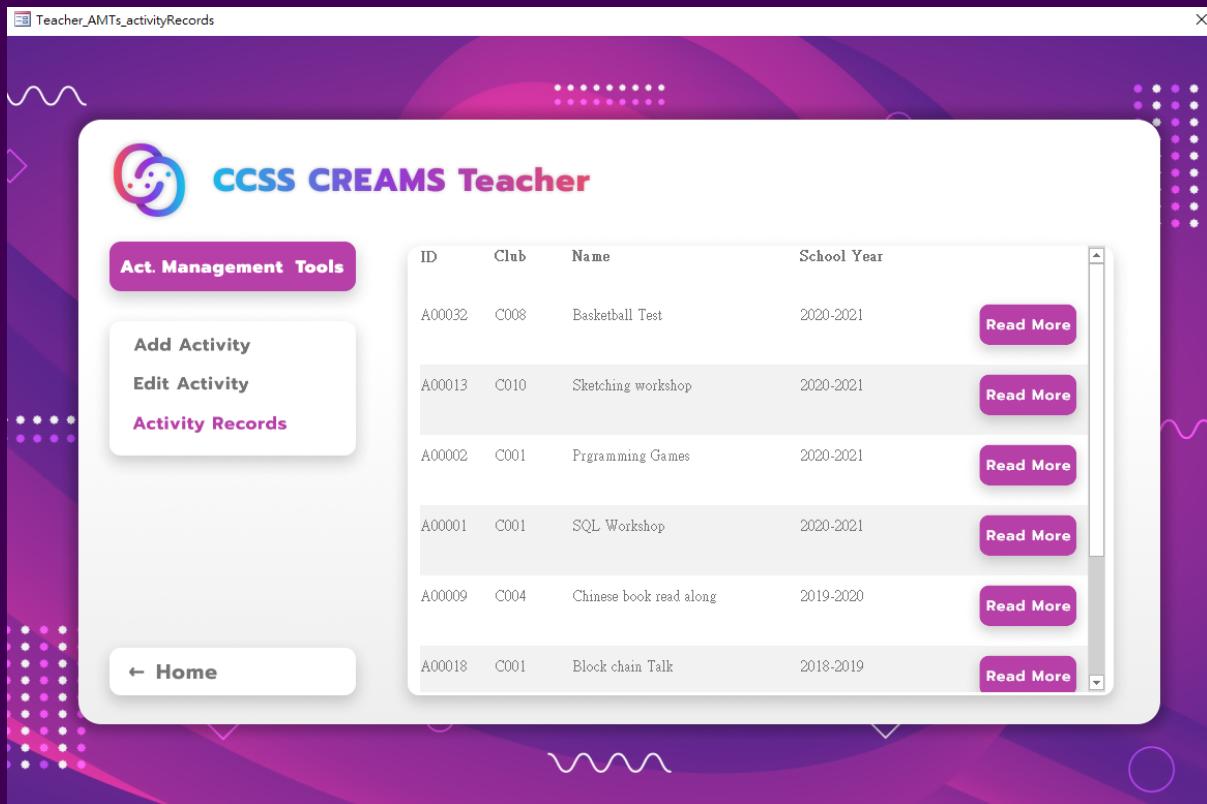
```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Teacher_AMTs_editActivity"
End Sub

Private Sub Form_Load()
    Me.txt_actid.Caption = getCurrentTMPid2
    Me.Refresh
    'DoCmd.Requery
End Sub
```

The last function of ATMs is Activity Records which leads the teacher to Teacher_ATMs_activityRecords forms that shows **the past activities that the particular teacher was in charge of.**



VBA of Teacher_ATMs_activityRecords

Option Compare Database
Option Explicit

```
Private Sub btn_HomeLogoOverlay_Click()

    BackToTeacherHome

End Sub

Private Sub btn_HomeTextOverlay_Click()

    BackToTeacherHome

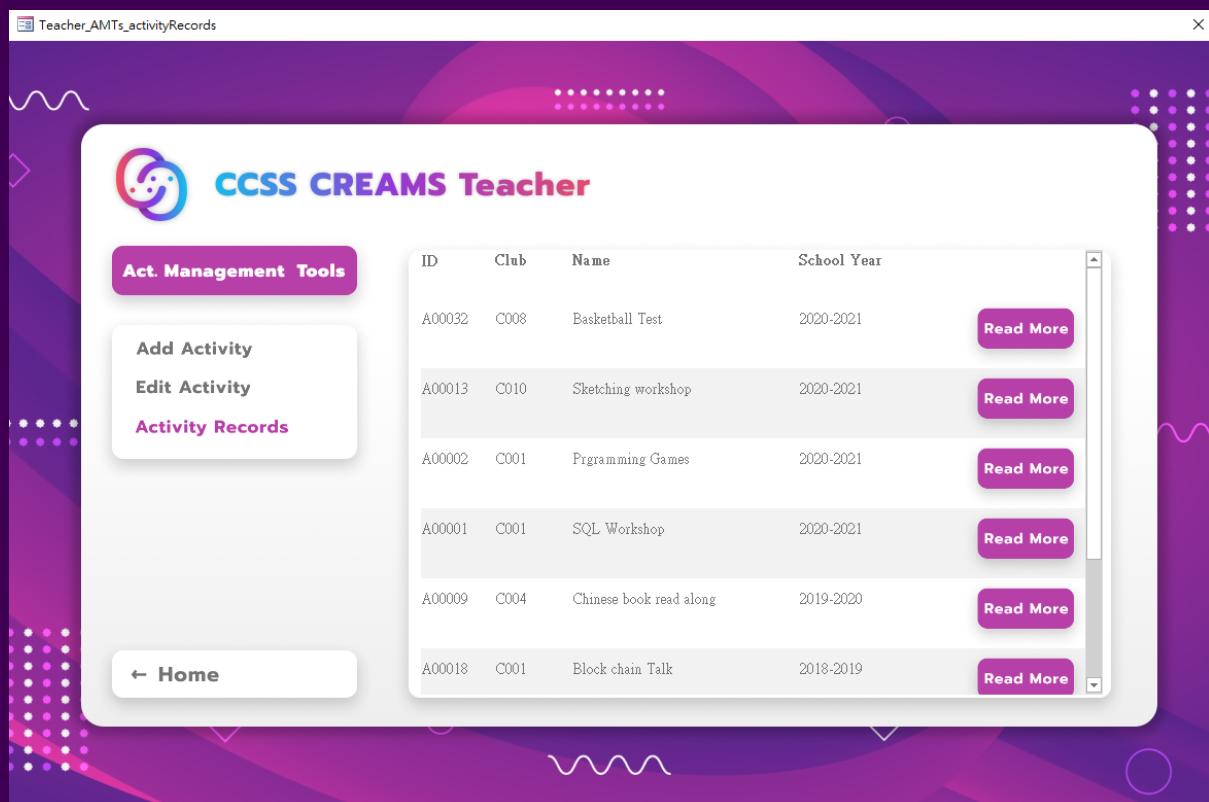
End Sub
Private Sub btn_HomeOverlay_Click()

    BackToTeacherHome

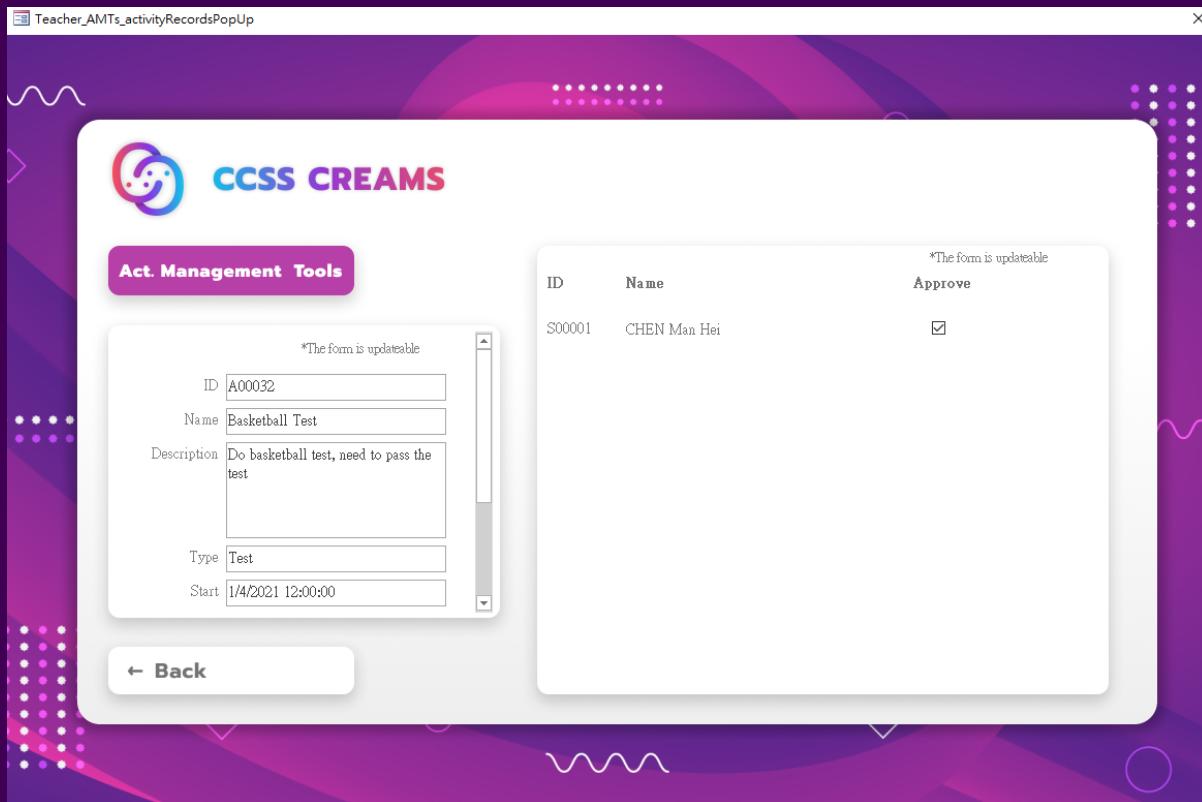
End Sub

Private Sub btn_nav1Overlay_Click()
```

```
ToTempPage "Teacher_AMTs_addActivity"  
End Sub  
  
Private Sub btn_nav2Overlay_Click()  
    ToTempPage "Teacher_AMTs_editActivity"  
End Sub  
  
Private Sub btn_nav3Overlay_Click()  
    ToTempPage "Teacher_AMTs_activityRecords"  
End Sub  
  
Private Sub Form_Load()  
  
    Me.txt_clubid.Caption = getCurrentTMPid  
    Me.txt_teacherid.Caption = getCurrentid  
    Me.txt_nowDate.Caption = getNowDate  
    Me.txt_nowTime.Caption = getNowTime  
  
    Me.Refresh  
    'DoCmd.Requery  
End Sub
```



The teacher can modify it when they press Read More when clicking the **Read More** button and the Teacher_ATMs_activityRecordsPopUp form shows up.



VBA of Teacher_ATMs_activityRecordsPopUp

```
Option Compare Database
Option Explicit
```

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeOverlay_Click()
    ToTempPage "Teacher_ATMs_activityRecords"
End Sub

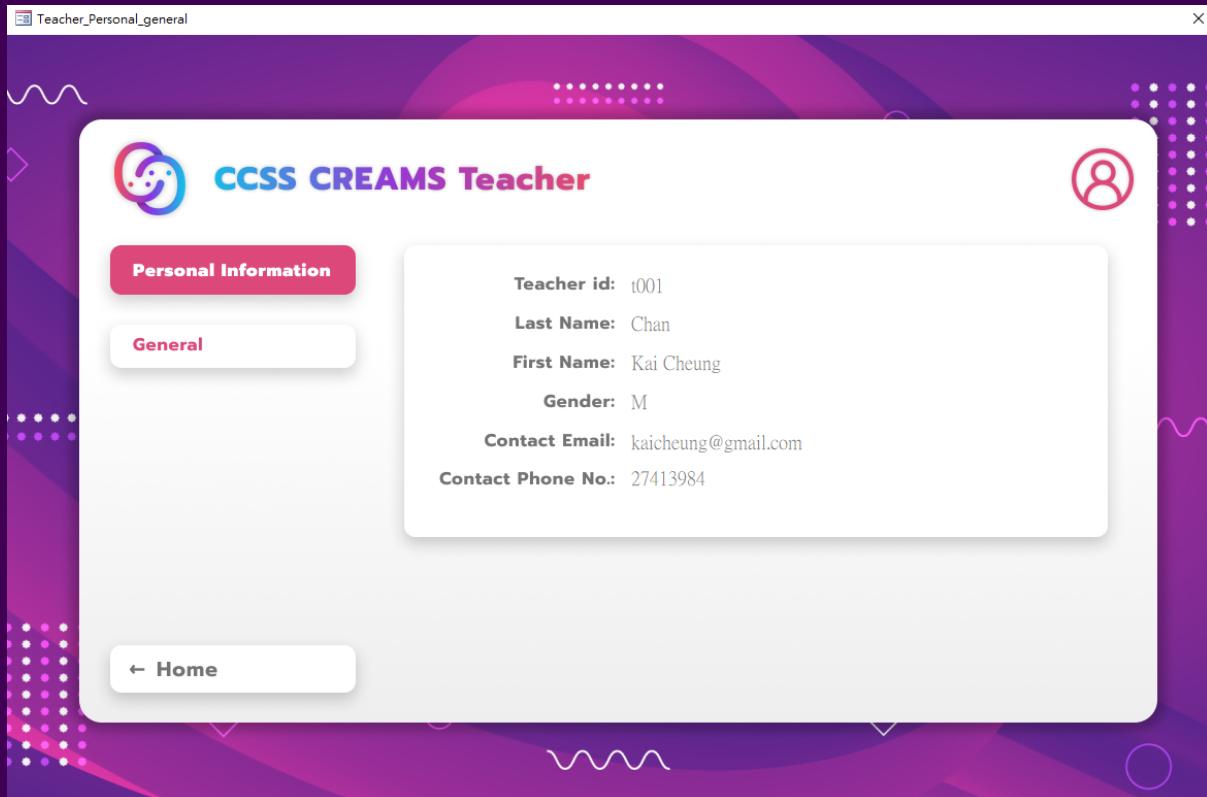
Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub Form_Load()
    Me.txt_clubid.Caption = getCurrentTMPid

```

```
Me.txt_schoolyear.Caption = getCurrentSchoolYear  
Me.txt_nowDate.Caption = getNowDate  
Me.txt_nowTime.Caption = getNowTime  
Me.txt_actid.Caption = getCurrentTMPid2  
  
Me.Refresh  
DoCmd.Requery  
  
End Sub
```

The Last one is the personal information part. After the teacher presses the circle in the top right corner in the teacher_welcome form, it will lead to **teachers' personal information**.



VBA of Teacher_Personal_general

Option Compare Database

```
Private Sub btn_HomeLogoOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_HomeTextOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_LogoutOverlay_Click()
    BackToTeacherHome
End Sub

Private Sub btn_PersonOverlay_Click()
    ToPage "Teacher_Personal_general"
```

```
End Sub

Private Sub Form_Load()

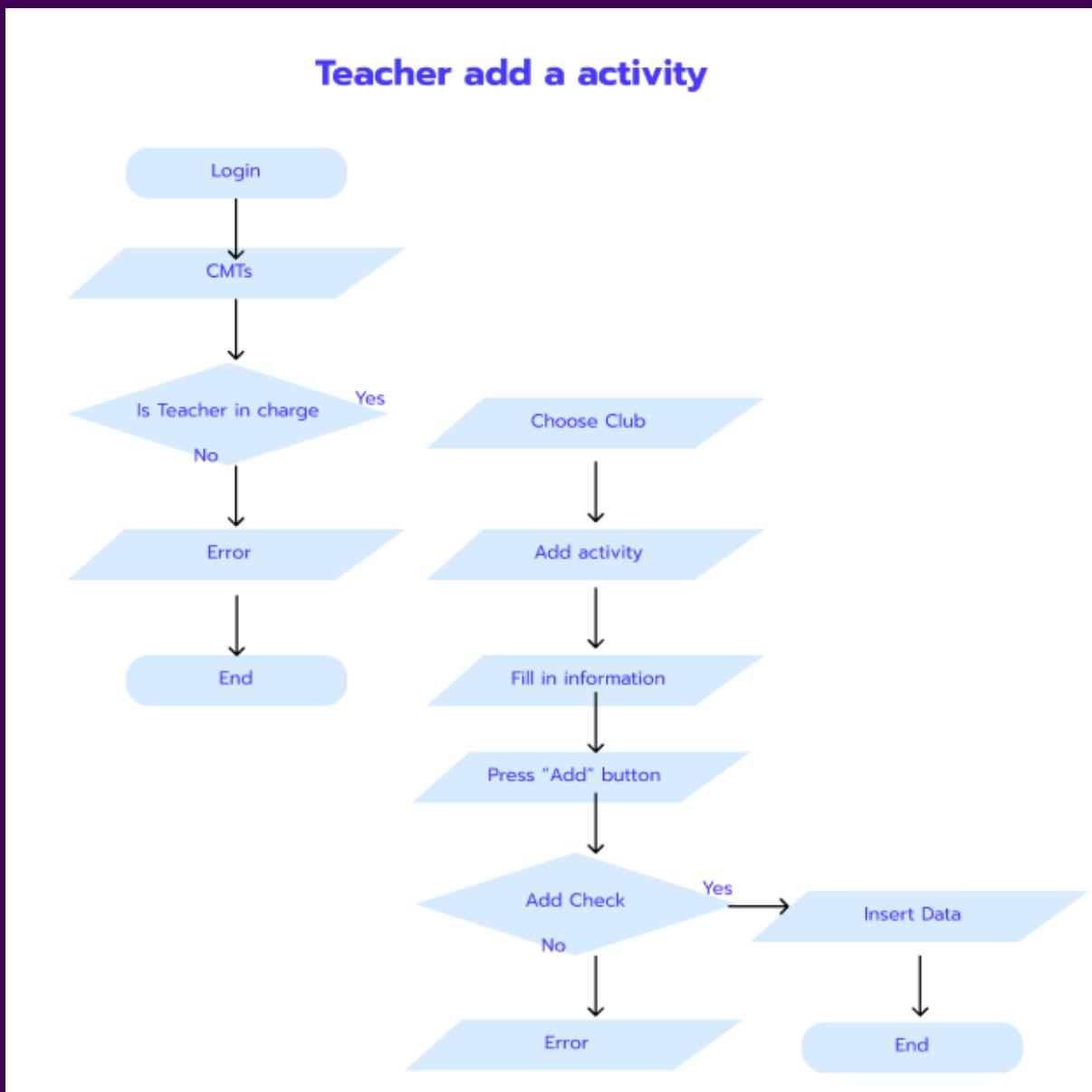
    Me.teacher_id.Caption = getCurrentid
    Me.Teacher_Lname.Caption = Nz(DLookup("[Teacher_lastName]", "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'", 0))
    Me.Teacher_Fname.Caption = Nz(DLookup("[Teacher(firstName]", "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'", 0))

    Me.Teacher_gender.Caption = Nz(DLookup("[Teacher_gender]", "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'", 0))

    Me.Teacher_email.Caption = Nz(DLookup("[Teacher_email]", "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'", 0))
    Me.Teacher_phoneNumber.Caption = Nz(DLookup("[Teacher_phoneNumber]", "Teacher_information", "[Teacher_id] = '" & [getCurrentid] & "'", 0))

End Sub
```

This leads to a teacher function which is add activity. The Flow of it is:



6. Logic design (Reports)

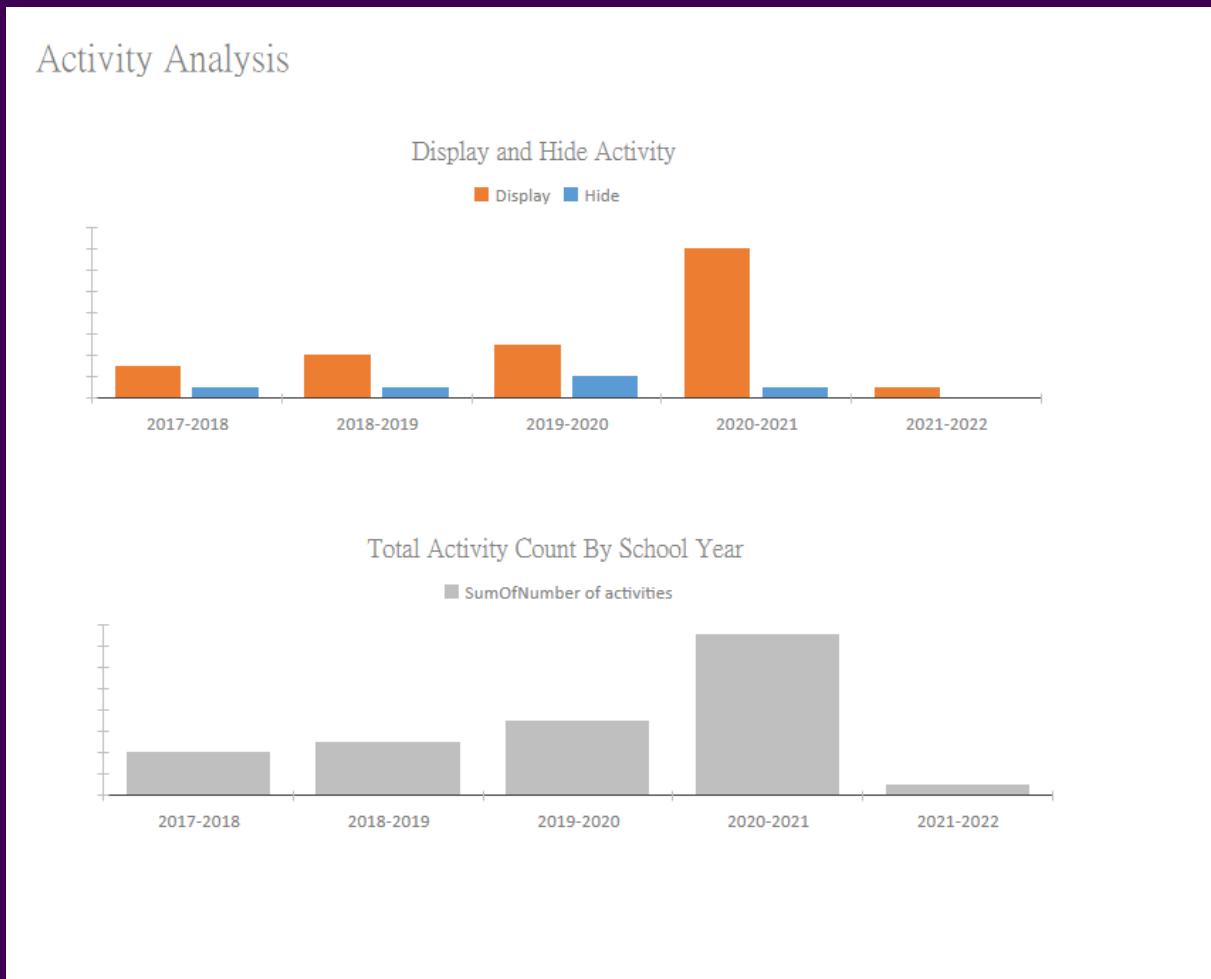
1. The First Report Activities_activityAttendanceRate which shows the attendance rate of each activities and show the attendance rate of all activities



2. The second report is Activities_enrollmentRate which shows the enrollment rates of each activity and the enrollment rate of all activities.



3. The third report is Activity_Analysis which shows the number of display and not display activities and groups by school year. Also, it will display the total activity in each school year.



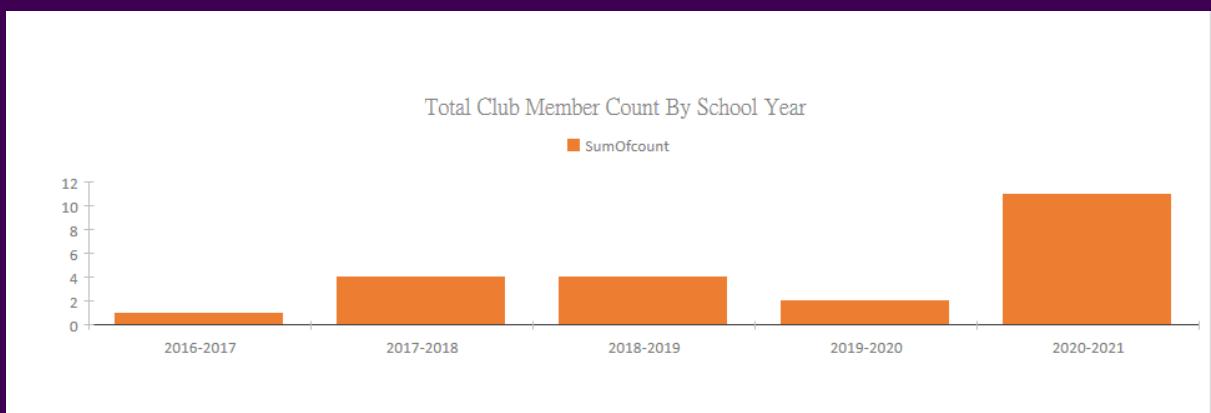
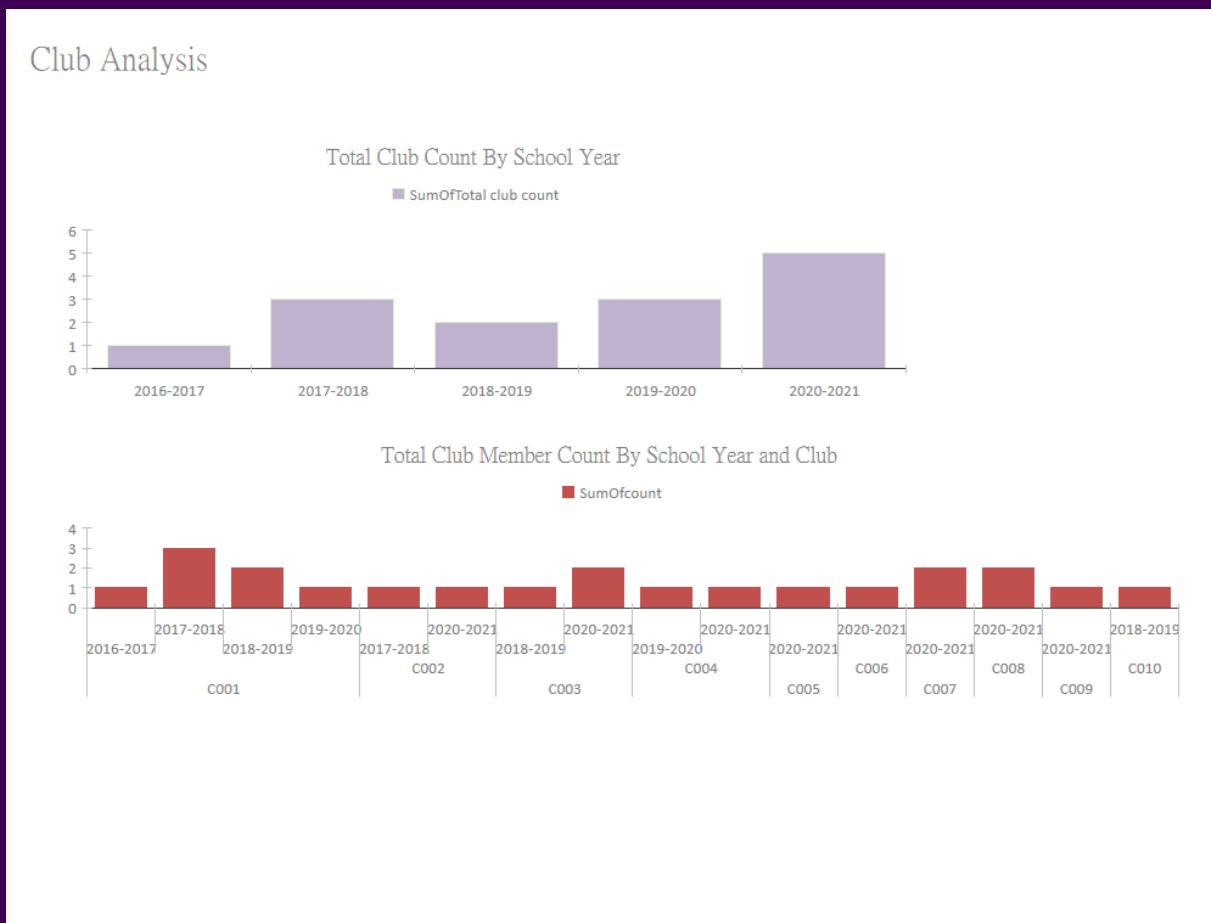
4. The next report is Activity_Analysis_2020-2021 which shows the percentage of hide and display activities in 2020-2021. And this is going to compare the number of activities in 2020-2021 and the number of activities last year which is 2019-2020.



5. The fifth report is ChairpersonOfClub_2020-2021 which shows all clubs that are still in operation and it's chairman's information.

Query_chairpersonOfClub_2020-2021						2021年4月16日 1:16:48
schoolYear	student_id	student_firstname	Student_lastname	Student_email	Club_id	
2020-2021	S00006	Tsz Ching	NG	TszChing@gmail.com	C007	
2020-2021	S00012	Yau Nam	NG	YauNam@gmail.com	C002	
2020-2021	S00013	Cheuk Wang	CHAN	CheukWang@gmail.com	C003	
2020-2021	S00015	Ming Chit	HO	MingChit@gmail.com	C004	
2020-2021	S00016	Hoi Lam	LEE	HoiLam@gmail.com	C005	
2020-2021	S00017	Wing Lun	LO	WingLun@gmail.com	C006	
2020-2021	S00018	Wing Kit	WONG	WingKit@gmail.com	C007	
2020-2021	S00019	Shing Yin	CHO	ShingYin@gmail.com	C008	
2020-2021	S00020	Tak Ming	CHAN	TakMing@gmail.com	C009	
2020-2021	S00001	Man Hei	CHEN	ManHei@gmail.com	C008	
2020-2021	S00001	Man Hei	CHEN	ManHei@gmail.com	C003	

6. This sixth report is Club_Analysis which has three charts. The first chart is the total club count by school year. The second chart is the total member count by school year and club. The third chart is the total club member grouping by school year.



7. The seventh report is Club_Analysis_2020-2021 which the first chart shows the comparison of the number of clubs in 2020-2021 and that in 2019-2020.



8. The eighth report is ClubList_2020-2021 which shows all club information which are operating in the school year 2020-2021.

ClubList_2020-2021

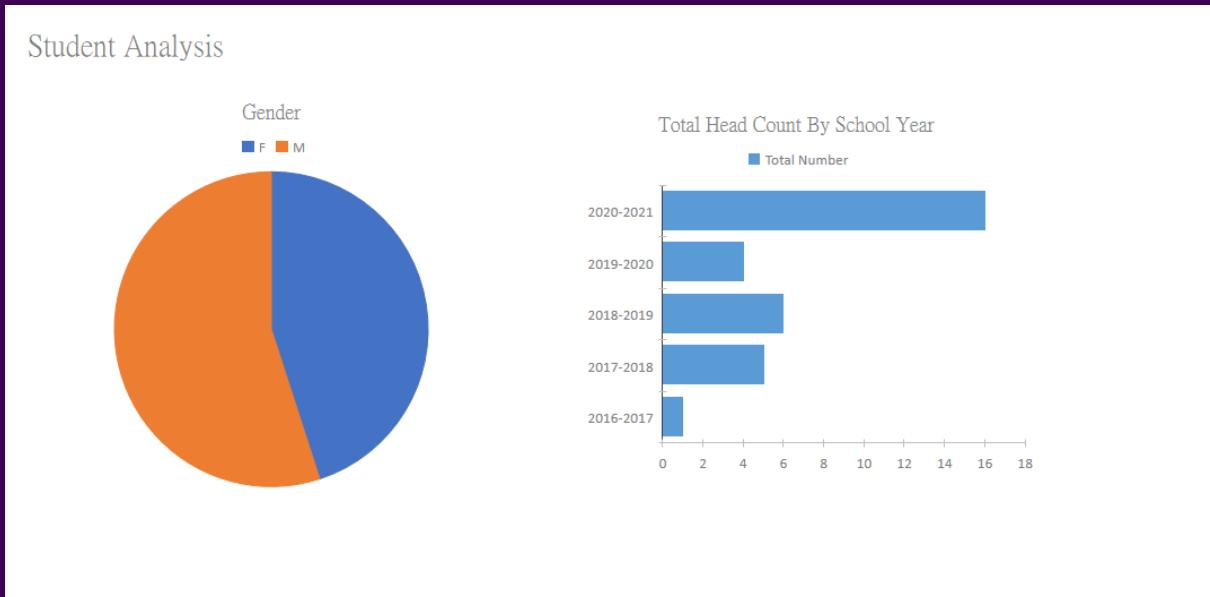
2021年4月16日
1:44:12

club_id	club_name	club_description	club_pending
C003	Math Club	Learn Math	<input checked="" type="checkbox"/>
C007	Tennis Club	Play tennis	<input checked="" type="checkbox"/>
C008	Basketball Club	Play basketball	<input checked="" type="checkbox"/>
C005	Drama Club	Learn Drama	<input checked="" type="checkbox"/>
C009	Music Club	Learn music	<input checked="" type="checkbox"/>

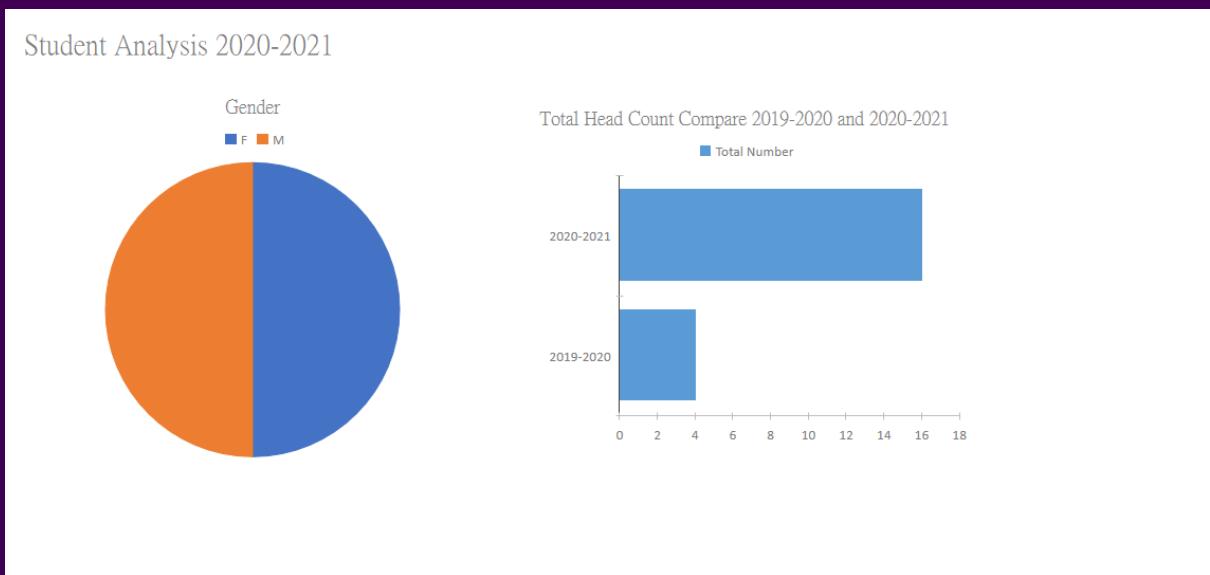
5

Page 1 of 1

9. The ninth report is Student_Analysis which one chart shows the ratio of the female and male in the CC Secondary School. The other chart is to show the total students grouping by school year.



10. The tenth report is Student_Analysis_2020-2021 which shows the ratio of the gender who are studied in cc secondary school in school year 2020-2021 and the total student compared to last school year.



11. The eleventh report, which is called Student_List, shows all the student(including graduated) student ID, First Name, Last Name and Gender.

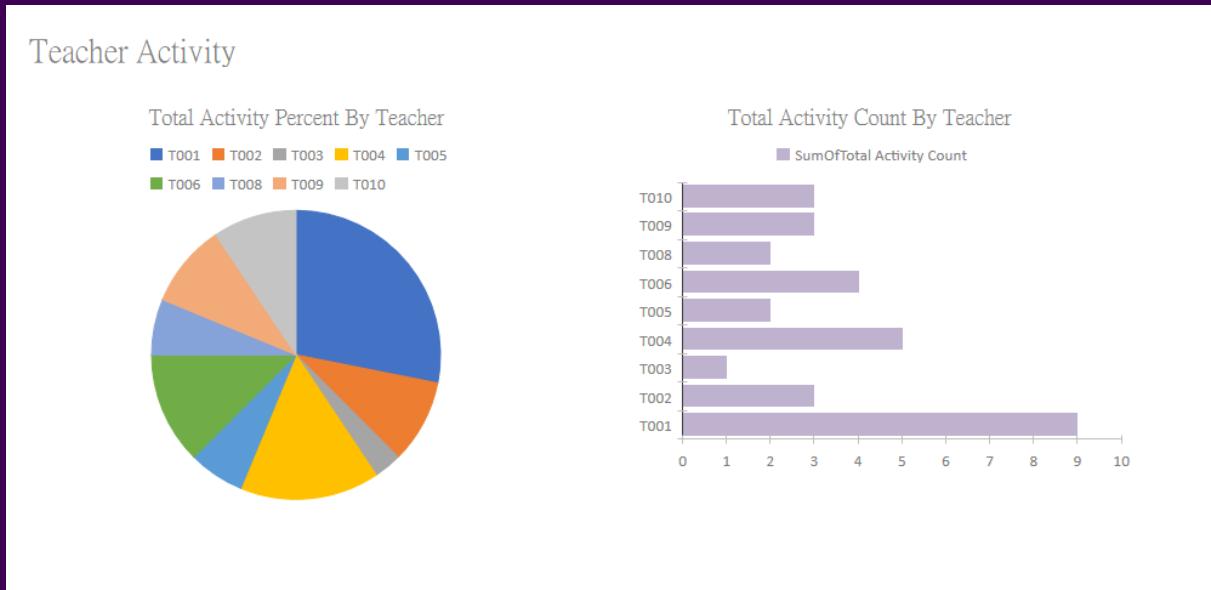
Student List of All Time

ID	First Name	Last Name	Gender
S00001	Man Hei	CHEN	M
S00002	Chun Ting	CHEUNG	M
S00003	Tsz Lung	CHAN	M
S00004	Ching Him	TAM	M
S00005	Tsz Yi	CHAN	F
S00006	Tsz Ching	NG	F
S00007	Pak Lun	LAI	M
S00008	Ho Ming	MOK	M
S00009	Tak Ming	CHAN	M
S00010	Chun Hei	HO	M
S00011	Ching Moon	CHOW	F
S00012	Yau Nam	NG	F
S00013	Cheuk Wang	CHAN	F
S00014	Wing Man	SHAM	F
S00015	Ming Chit	HO	M
S00016	Hoi Lam	LEE	F
S00017	Wing Lun	LO	M
S00018	Wing Kit	WONG	F
S00019	Shing Yin	CHO	M
S00020	Tak Ming	CHAN	F

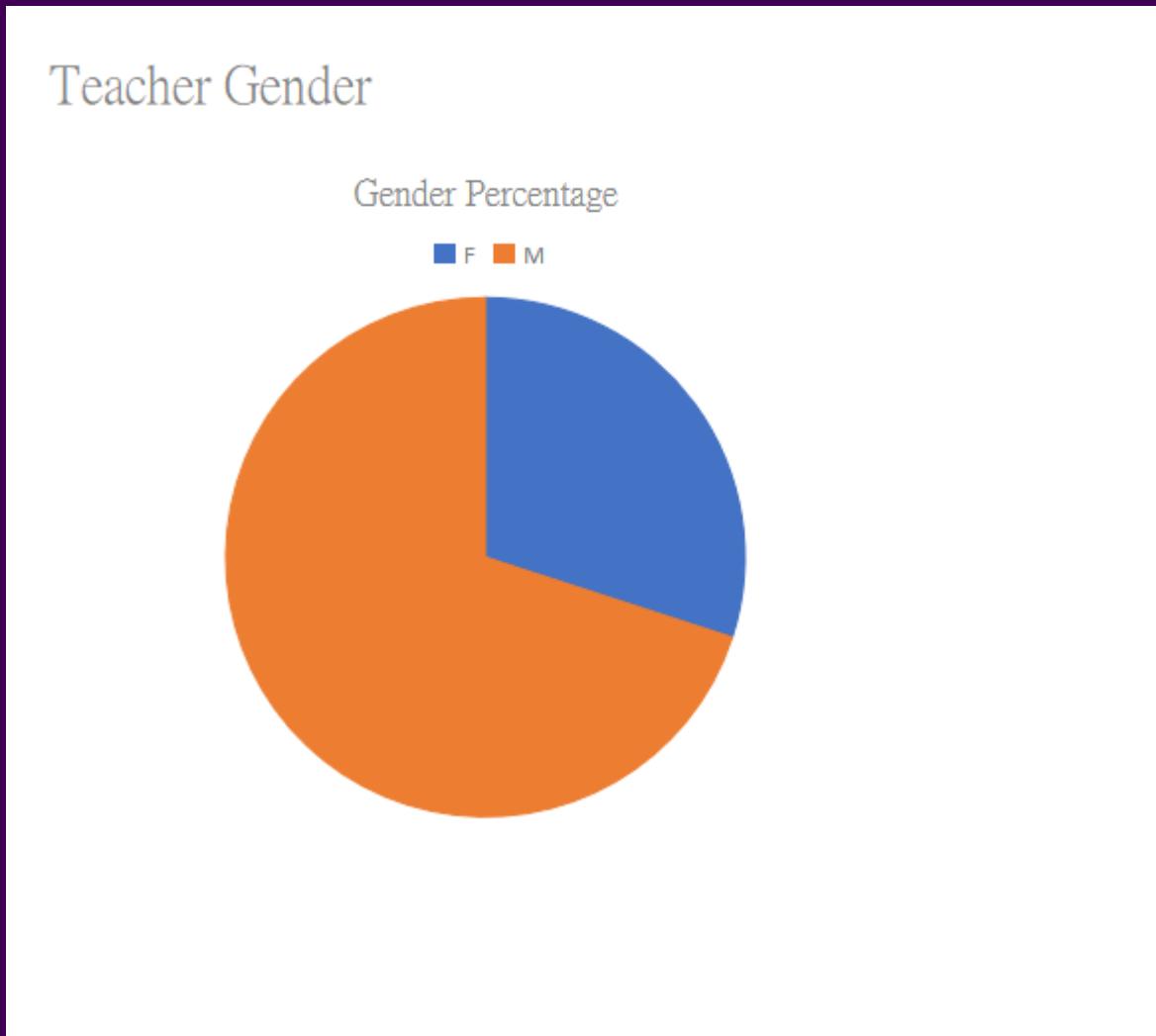
12. The twelfth report is StudentList_2020-2021 and it shows all student grade, class, class number along with student ID which was studied in CC Secondary School in School Year 2020-2021.

StudentList_2020-2021					2021年4月16日	1:57:13
Student_id	SchoolYear	Student_grade	Student_class	Student_classNumber		
S00006	2020-2021	4	E	9		
S00007	2020-2021	2	C	3		
S00008	2020-2021	5	C	16		
S00009	2020-2021	5	D	21		
S00010	2020-2021	2	C	11		
S00011	2020-2021	1	E	2		
S00012	2020-2021	2	E	28		
S00013	2020-2021	6	E	8		
S00014	2020-2021	5	E	16		
S00015	2020-2021	3	F	29		
S00016	2020-2021	5	F	18		
S00017	2020-2021	2	C	30		
S00018	2020-2021	3	A	22		
S00019	2020-2021	6	C	2		
S00020	2020-2021	4	A	9		
S00001	2020-2021	6	A	23		

13. The thirteenth report is Teacher_Activity which shows each teachers' number of activity in charge percentage. And the other chart is the total activity count that each teacher is in charge of grouping by teacher.



14. The fourteenth report is Teacher_Gender which shows the percentage of teachers' gender.



15. The fifteenth report is Teacher_List which shows the list of Teachers with their teacher ID, last name, first name and Gender.

Teacher List of All Time			
ID	Lastname	Teacher_firstname	Gender
T001	Chan	Kai Cheung	M
T002	Lam	Tai Man	M
T003	Wong	Hoi Ting	F
T004	Zhong	Sum Lam	M
T005	Deng	Chong Sum	M
T006	Ho	Mau Hang	M
T007	Zheng	Sum Yu	F
T008	Cheung	Sai Wan	M
T009	Mang	Billy	M
T010	Shum	Bo Lan	F

2021年4月16日

Page 1 of 1

7. Logic design (Queries)

There are two types of queries in this database **Search Queries and Insert Queries.**

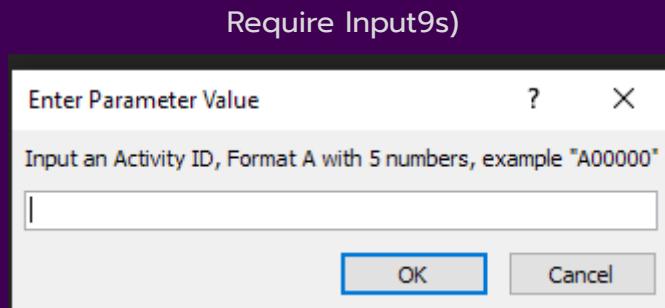
There are two types of queries in this database. **Normal and Linked Queries.**

The first one is Normal queries. It works like normal queries. **Users can simply click** on them to use. Queries under this type might require user inputs.

The name of this type of query starts with "Query_". The format is "Query_[Output](_WITH_[Field])(_BY_[Field])"

The field(s) followed by "WITH_" means that the user has to input that information to run the query.

The field(s) followed by "BY_" means that the outputs are grouped by that field.



List of example queries

-  [Query_studentGender_2020-2021](#)
-  [Query_studentGender_BY_gender](#)
-  [Query_studentGradeInformation_WITH_studentID](#)
-  [Query_studentList_2020-2021](#)
-  [Query_studentList_WITH_schoolYear](#)
-  [Query_teacherGender_BY_gender](#)
-  [Query_teacherInforamtion_BY_activityID](#)
-  [Query_teacherInforamtion_WITH_clubID&SchoolYear](#)
-  [Query_teacherInformation_BY_clubID&SchoolYear](#)
-  [Query_totalActivityCount_19-20_20-21Compare](#)
-  [Query_totalActivityCount_BY_schoolYear](#)
-  [Query_totalActivityCount_BY_teacher](#)
-  [Query_totalClubCount_19-20_20-21Compare](#)

The other type of queries is "QueryLink". This type of query can be opened by click, however it might not work for user inputs, since these queries are made/ designed as datasource use by forms. These queries will auto get the quired fields/ records from the form. Therefore, it is only designed to open this type of queries using forms, instead of running it without forms.

This type of queries will contain "QueryLink" in the second word of their name.

Usually this type of queries are link the a subform named "Querysubform".

For example, "Student_QueryLink_selfAccount". This means the query is used by Form "Student_selfAccount" and this query is linked to the form "Student_Querysubform_selfAccount".

List of example queries

Student_QueryLink_academicRecord
Student_QueryLink_applicationStatus_act
Student_QueryLink_applicationStatus_club
Student_QueryLink_awardsRecord
Student_QueryLink_browseActivities
Student_QueryLink_browseClubs
Student_QueryLink_CMTs_attendanceOverview
Student_QueryLink_CMTs_attendancePopUp
Student_QueryLink_CMTs_popUP
Student_QueryLink_CMTs_records
Student_QueryLink_CMTs_recordsPopUp_left
Student_QueryLink_CMTs_recordsPopUp_right
Student_QueryLink_records_act
Student_QueryLink_records_club
Student_QueryLink_selfAccount

Student_QueryLink_records_act
Student_QueryLink_records_club
Student_QueryLink_selfAccount
Student_Personal_medicalInformation
Student_Personal_selfAccount
Student_Querysubform_academicRecord
Student_Querysubform_ApplicationStatus_act
Student_Querysubform_CMTs_recordsPopUp_right
Student_Querysubform_Records_act
Student_Querysubform_Records_club
Student_Querysubform_selfAccount

(a) Normal Search Queries

We will focus on talking about Normal Queries first.

Query_activityAttendanceRate

Query_activityAttendanceRate_BY_activity

This two queries are used for query activity attendance rate, the second one will group by activity

SQL

```
SELECT ((select count(Activity_record_attendance)
from Activity_record
where Activity_record_attendance = TRUE)/count(*)*100) AS [Whole
Percentage]
FROM Activity_record;
```

SQL

```
SELECT Activity_id, ((select count(Activity_record_attendance)
from Activity_record as c
where Activity_record_attendance = TRUE and
c.Activity_id=a.Activity_id)/count(*)*100) AS [Whole Percentage]
FROM Activity_record AS a
GROUP BY Activity_id;
```

Query_activityEnrollmentRate

Query_activityEnrollmentRate_BY_activity

This two queries are used for query activity Enrollment rate, the second one will group by

SQL

```
SELECT (Count(*)/sum(Activity_maxParticipants)*100) AS [Whole
Percentage]
FROM Activity_information, Activity_record
WHERE
Activity_information.Activity_id=Activity_record.Activity_id;
```

SQL

```
SELECT Activity_information.Activity_id,
```

```
(Count(student_id) / sum(Activity_maxParticipants) *100) AS
Percentage
FROM Activity_information, Activity_record
WHERE
Activity_information.Activity_id=Activity_record.Activity_id
GROUP BY Activity_information.Activity_id;
```

Query_activityList_WITH_schoolYear

This queries is used for query activity list with school year input

SQL

```
SELECT *
FROM Activity_information
WHERE SchoolYear=[Input a school year, example "2020-2021"];
```

Query_activityRecords_WITH_activityID

This one query is used for query activity records with activity id input

SQL

```
SELECT *
FROM Activity_record
WHERE Activity_ID=[Input an Activity ID, Format A with 5 numbers,
example "A0000"];
```

Query_chairpersonOfClub_2020-2021

This one query is used for query chairperson of club in 2020-2021

SQL

```
SELECT Club_record.schoolYear, Student_information.student_id,
Student_information.student_firstname,
Student_information.Student_lastname,
Student_information.Student_email, Club_record.Club_id
FROM Student_information, Club_record
WHERE ((Club_record.schoolYear)="2020-2021") AND
((Student_information.student_id)=[Club_record].[student_id]));
```


Query_classHeadCount_WITH_schoolYear&Grade

This query is used for query class headcount, need to input school year and grade

SQL

```
SELECT Count(*) AS [number of students in the class]
FROM student_grade
WHERE ((student_grade.schoolYear)=[Input a school year, example
"2020-2021"]) And ((student_grade.student_grade)=[Input the
student's grade, example "1"])
GROUP BY student_grade.student_class;
```

*Query_clubList_2020-2021**Query_clubList_WITH_schoolYear**Query_clubList_WITH_StudentID*

This two queries are used for query club list, first one are records in 2020-2021, second need input school year, third need input student id

SQL

```
SELECT a.club_id, a.club_name, a.club_description, a.club_pending
FROM club_information AS a, Club_schoolyear AS b
WHERE a.club_id = b.club_id AND
b.schoolYear="2020-2021";
```

SQL

```
SELECT a.club_id, a.club_name, a.club_description, a.club_pending
FROM club_information AS a, Club_schoolyear AS b
WHERE a.club_id = b.club_id AND
b.schoolYear=[Input a year, example "2020-2021"];
```

SQL

```
SELECT a.club_id, a.club_name, a.club_description, a.club_pending
FROM club_information AS a, Club_schoolyear AS b
WHERE a.club_id=b.club_id
and STUDENT_ID=[Input a student_id, example: "S00001"];
```

Query_clubRecords_WITH_clubID

Query_clubRecords_WITH_clubID&SchoolYear

This two queries are used for query club records, first one need input club id , second need input club id and school year

SQL

```
SELECT *
FROM club_record
WHERE club_id =[Input a club id, example:"C001"];
```

SQL

```
SELECT *
FROM CLUB_RECORD
WHERE club_id=[Input a club id, example: "C001"] and
schoolyear=[Input a school year, example: "2020-2021"];
```

Query_displayAndNotDisplayActivity_2020-2021

Query_displayAndNotDisplayActivity_schoolYear

This two queries are used for query display and not display activity in 2020-2021 and the other is group by school year

SQL

```

SELECT DISTINCT c.Display, c.Hide
FROM Activity_information AS d LEFT JOIN (SELECT
a.SchoolYear,Show AS Display, Not_Show AS Hide
    FROM(SELECT SchoolYear , COUNT(Activity_Name) AS [Show]
    FROM Activity_information
    WHERE Activity_display = TRUE
    GROUP BY SchoolYear
    ORDER BY SchoolYear) as a
LEFT JOIN(SELECT SchoolYear , COUNT(Activity_Name) AS [Not_Show]
    FROM Activity_information
    WHERE Activity_display = FALSE
    GROUP BY SchoolYear
    ORDER BY SchoolYear) as b ON b.SchoolYear= a.SchoolYear
WHERE a.SchoolYear IS NOT NULL
UNION
SELECT a.SchoolYear,Show AS Display, Not_Show AS Hide
    FROM(SELECT SchoolYear , COUNT(Activity_Name) AS [Not_Show]
    FROM Activity_information
    WHERE Activity_display = FALSE
    GROUP BY SchoolYear
    ORDER BY SchoolYear) as a
LEFT JOIN(SELECT SchoolYear , COUNT(Activity_Name) AS [Show]
    FROM Activity_information
    WHERE Activity_display = TRUE
    GROUP BY SchoolYear
    ORDER BY SchoolYear) as b ON b.SchoolYear= a.SchoolYear
WHERE a.SchoolYear IS NOT NULL) AS c ON d.SchoolYear =
c.SchoolYear
WHERE (((d.SchoolYear)="2020-2021"));

```

SQL

```

SELECT DISTINCT d.SchoolYear, Display, Hide
FROM Activity_information AS d LEFT JOIN (SELECT
a.SchoolYear,Show AS Display, Not_Show AS Hide
    FROM(SELECT SchoolYear , COUNT(Activity_Name) AS [Show]
    FROM Activity_information
    WHERE Activity_display = TRUE
    GROUP BY SchoolYear
    ORDER BY SchoolYear) as a
LEFT JOIN(SELECT SchoolYear , COUNT(Activity_Name) AS [Not_Show]
    FROM Activity_information
    WHERE Activity_display = FALSE
    GROUP BY SchoolYear
    ORDER BY SchoolYear) as b ON b.SchoolYear= a.SchoolYear
WHERE (((d.SchoolYear)="2020-2021"));

```

```
GROUP BY SchoolYear  
ORDER BY SchoolYear) as b ON b.SchoolYear= a.SchoolYear  
WHERE a.SchoolYear IS NOT NULL  
UNION  
SELECT a.SchoolYear, Show AS Display, Not_Show AS Hide  
FROM(SELECT SchoolYear , COUNT(Activity_Name) AS [Not_Show]  
FROM Activity_information  
WHERE Activity_display = FALSE  
GROUP BY SchoolYear  
ORDER BY SchoolYear) as a  
LEFT JOIN(SELECT SchoolYear , COUNT(Activity_Name) AS [Show]  
FROM Activity_information  
WHERE Activity_display = TRUE  
GROUP BY SchoolYear  
ORDER BY SchoolYear) as b ON b.SchoolYear= a.SchoolYear  
WHERE a.SchoolYear IS NOT NULL) AS c ON c.SchoolYear =  
d.SchoolYear;
```

Query_gradeHeadCount_WITH_schoolYear

This two queries are used for query students grade headcount, need to input school year

SQL

```
SELECT student_grade AS [student grade], count (*) AS [number of  
students in the grade]  
FROM student_grade  
WHERE schoolYear=[Input a school year, example "2020-2021"]  
GROUP BY student_grade;
```

Query_notDisplayActivity_BY_schoolYear

This query is used for query activity that are not displaying by school year

SQL

```
SELECT SchoolYear, COUNT(*)
FROM Activity_information
WHERE Activity_display = FALSE
GROUP BY SchoolYear
ORDER BY SchoolYear;
```

Query_studentClassNameList_WITH_schoolYear

This query are used for query student name list, need input school year

SQL

```
SELECT student_firstname, student_lastname, student_grade,
student_class
FROM student_information AS i, student_grade AS g
WHERE i.student_id = g.student_id
and schoolYear=[Input a school year, example "2020-2021"];
```

*Query_studentGender_2020-2021**Query_studentGender_BY_gender**Query_studentGradeInformation_WITH_studentID*

These three queries are used for query student Gender, the first one is in 2020-2021, the second one will group by gender, the third need to input student id

SQL

```
SELECT Student_gender, count(*) AS [Count]
FROM student_information, Student_grade
WHERE student_information.student_id=Student_grade.student_id
    AND SchoolYear="2020-2021"
GROUP BY student_gender;
```

SQL

```
SELECT Student_gender, count(*) AS [Count]
FROM student_information
GROUP BY student_gender;
```

SQL

```
SELECT student_id, schoolyear, student_grade, student_class
FROM student_grade
WHERE student_id = [Input a student id, example "S00001"];
```

*Query_studentList_2020-2021**Query_studentList_WITH_schoolYear*

This two queries are used for query student list in 2020-2021, the second one require user in input school year

SQL

```
SELECT *
FROM Student_grade
WHERE schoolYear="2020-2021";
```

SQL

```
SELECT *
FROM Student_information, Student_grade
WHERE SchoolYear=[Input a school year, example: "2020-2021"]
      AND
Student_information.student_id=Student_grade.Student_id;
```

Query_teacherGender_BY_gender

This query are used for query teacher total gender by gender

SQL

```
SELECT Teacher_gender, count(*) AS [Count]
FROM Teacher_information
GROUP BY Teacher_gender;
```


Query_teacherInformation_BY_activityID

Query_teacherInformation_WITH_clubID&SchoolYear

Query_teacherInformation_BY_clubID&SchoolYear

These three queries are used for query teacher Information, the first one group by activity ID, the second one require user to input club id and school year

SQL

```
SELECT Activity_id, Teacher_information.*  
FROM Teacher_information, Activity_information  
WHERE Teacher_information.Teacher_id =  
Activity_information.Teacher_id  
ORDER BY Activity_startDate DESC , Activity_startTime;
```

SQL

```
SELECT club_id, schoolYear, Teacher_information.*  
FROM Teacher_information, Club_schoolyear  
WHERE Teacher_information.Teacher_id = Club_schoolyear.Teacher_id  
AND Club_id=[Input a club ID, example "C001"]  
AND schoolYear=[Input a school year, example "2020-2021"];
```

SQL

```
SELECT club_id, schoolYear, Teacher_information.*  
FROM Teacher_information, Club_schoolyear  
WHERE Teacher_information.Teacher_id = Club_schoolyear.Teacher_id  
ORDER BY SchoolYear DESC , Club_id;
```

Query_totalActivityCount_19-20_20-21Compare

Query_totalActivityCount_BY_schoolYear

Query_totalActivityCount_BY_teacher

These three queries are used for query total activity count, the first one will compare 19-20 and 20-21 the second one will group by school year, the third one group by teacher.

SQL

```
SELECT SchoolYear AS [School year], COUNT(*) AS [Number of activities]
FROM Activity_information
WHERE SchoolYear in ("2020-2021", "2019-2020")
GROUP BY SchoolYear
ORDER BY SchoolYear;
```

SQL

```
SELECT SchoolYear AS [School year], COUNT(*) AS [Number of activities]
FROM Activity_information
GROUP BY SchoolYear
ORDER BY SchoolYear;
```

SQL

```
SELECT Teacher_information.Teacher_id,
Count(Activity_information.Activity_id) AS [Total Activity Count]
FROM Activity_information, Teacher_information
WHERE
(((Teacher_information.Teacher_id)=[Activity_information].[Teacher_id]))
GROUP BY Teacher_information.Teacher_id;
```

Query_totalClubCount_19-20_20-21Compare

Query_totalClubCount_BY_schoolYear

These two queries are used for query total Club Count, the first one will compare 19-20 and 20-21. the second one will group by school year.

SQL

```
SELECT club_schoolyear.SchoolYear, Count(*) AS [Total club count]
FROM club_schoolyear
WHERE schoolyear in ("2020-2021","2019-2020")
GROUP BY club_schoolyear.SchoolYear;
```

SQL

```
SELECT club_schoolyear.SchoolYear, Count(*) AS [Total club count]
FROM club_schoolyear
GROUP BY club_schoolyear.SchoolYear;
```

Query_totalClubMemberHeadCount_BY_Club_2020-2021

Query_totalClubMemberHeadCount_BY_schoolYear

Query_totalClubMemberHeadCount_BY_schoolYear&Club

These three queries are used for query activity total club member headcount, the first one will only get records in 20-21, second one will group by school year, third one will group by school year and club

SQL

```
SELECT club_id, Count(*) AS [count]
FROM club_record
WHERE schoolyear ="2020-2021"
GROUP BY club_id;
```

SQL

```
SELECT schoolyear, Count(*) AS [count]
FROM club_record
GROUP BY schoolyear;
```

SQL

```
SELECT schoolyear, club_id, Count(*) AS [count]
FROM club_record
GROUP BY schoolyear, club_id;
```

Query_totalHeadCount_19-20_20-21Compare

Query_totalHeadCount_BY_schoolYear

These two queries are used for querying the total headcount of students, the first one compares between 19-20 and 20-21. The second one will group by school year.

SQL

```
SELECT student_grade.schoolyear, Count(*) AS [total number  
students in a school year]  
FROM student_grade  
WHERE (((student_grade.schoolyear) In ("2020-2021","2019-2020")))  
GROUP BY student_grade.schoolyear;
```

SQL

```
SELECT schoolyear, count (*) AS [total number students in a  
school year]  
FROM student_grade  
GROUP BY schoolyear;
```

Query_totalTeacherCount

This is a query to count total teacher.

SQL

```
SELECT Count(*) AS [Total Teacher Head Count]  
FROM Teacher_information;
```

(b) Linked Search Queries

As mentioned above, these queries are not meant to run on their own.

Student_QueryLink_academicRecord

This is a linked query for showing student academic record

SQL

```
SELECT Student_academic.SchoolYear, Student_academic.Student_sem,
Student_academic.Student_academic_chi,
Student_academic.Student_academic_eng,
Student_academic.Student_academic_maths,
Student_academic.Student_academic_ls
FROM Student_academic
WHERE
(((Student_academic.Student_id)=[Forms]![Student_Personal_academi
cRecords]![txt_stdid].[caption]));
```

Student_QueryLink_applicationStatus_act

Student_QueryLink_applicationStatus_club

This is a linked query for student application status activity and club

SQL

```
SELECT *
FROM Activity_record, Activity_information
WHERE
Activity_information.Activity_id=Activity_record.Activity_id
And Activity_record.student_id=
Forms![Student_applicationStatus]!txt_stdid.Caption
And
(Activity_endDate=Forms![Student_applicationStatus]!txt_nowDate.Caption
And
Activity_endTime>Forms![Student_applicationStatus]!txt_nowTime.Caption)
Or
(Activity_endDate>Forms![Student_applicationStatus]!txt_nowDate.Caption);
```

SQL

```
SELECT *
```

```

FROM Club_record, Club_information, Club_schoolyear
WHERE Club_information.Club_id=Club_record.Club_id
    AND Club_schoolyear.Club_id=Club_record.Club_id
    AND Club_schoolyear.SchoolYear = Club_record.SchoolYear
    And Club_record.student_id=
[Forms]![Student_applicationStatus]![txt_stdid].Caption
    And Club_record.SchoolYear =
[Forms]![Student_applicationStatus]![txt_schoolyear].caption;

```

Student_QueryLink_awardsRecord

This is a linked query for showing student awards record

SQL

```

SELECT *
FROM Student_awards
WHERE
Student_id=[Forms]![Student_Personal_awardsRecord]![txt_stdid].caption;

```

*Student_QueryLink_browseActivities**Student_QueryLink_browseClubs*

This is a linked query for showing student browse activities and clubs

SQL

```

SELECT Activity_startDate & " " & Activity_startTime AS Start,
Activity_endDate & " " & Activity_endTime AS [End],
Activity_registrationDeadline & " " &
Activity_registrationDeadlineTime AS [Registration Deadline], *
FROM Activity_information
WHERE ((Activity_registrationDeadline =
[Forms]![Student_browseActivities]![txt_nowDate].caption
    AND Activity_registrationDeadlineTime >
[Forms]![Student_browseActivities]![txt_nowTime].caption)
    OR (Activity_registrationDeadline >
[Forms]![Student_browseActivities]![txt_nowDate].caption))
    AND Activity_display = TRUE
    AND ((Club_id IN
        (SELECT Club_id
        FROM Club_record
        WHERE Student_id =
[Forms]![Student_browseActivities]![txt_stdid].caption
        AND SchoolYear =

```

```
[Forms]![Student_browseActivities]![txt_schoolyear].caption
    AND Club_approval = TRUE))
OR
    (Club_id IS NULL))
AND Activity_id NOT IN
    (SELECT Activity_information.Activity_id
     FROM Activity_record, Activity_information
      WHERE Activity_information.Activity_id =
Activity_record.Activity_id
          AND Activity_approval = TRUE
      GROUP BY
Activity_information.Activity_id,Activity_maxParticipants
          HAVING COUNT(Activity_record.Activity_id) >=
Activity_maxParticipants)
ORDER BY Activity_registrationDeadline,
Activity_registrationDeadlineTime,
Activity_information.Activity_id;
```

SQL

```
SELECT *
FROM Club_information, Club_schoolyear, Student_information,
Teacher_information
WHERE Club_information.Club_id=Club_schoolyear.Club_id
    AND
Club_schoolyear.Student_id=Student_information.Student_id
    AND
Club_schoolyear.Teacher_id=Teacher_information.Teacher_id
    AND
Club_schoolyear.SchoolYear=[Forms]![Student_browseClubs]![txt_sch
oolyear].caption
    AND Club_information.Club_id NOT IN
        (SELECT Club_information.Club_id
         FROM Club_record, Club_information
          WHERE Club_information.Club_id = Club_record.Club_id
              AND Club_approval = TRUE
          GROUP BY Club_information.Club_id,Club_maxParticipants
              HAVING COUNT(Club_record.club_id) >=
Club_maxParticipants);
```

Student_QueryLink_CMTs_attendanceOverview

Student_QueryLink_CMTs_attendancePopUp

This is a linked query for showing student CMTs attendance

SQL

```

SELECT DISTINCT Activity_information.Activity_id, activity_name,
Activity_endDate & " " & Activity_endTime AS [End]
FROM Activity_information, Activity_record
WHERE
Activity_information.Activity_id=Activity_record.Activity_id
    AND
Activity_information.Club_id=Forms! [Student_CMTs_attendance] !txt_
clubid.Caption
    AND Activity_display=TRUE
    AND Activity_approval=TRUE
    AND (
        ((Activity_startDate=Forms! [Student_CMTs_attendance] !txt_nowDate.
Caption
            And
Activity_startTime<Forms! [Student_CMTs_attendance] !txt_nowTime.Ca-
ption)
        Or
        (Activity_startDate<Forms! [Student_CMTs_attendance] !txt_nowDate.Ca-
ption))
    AND
        ((Activity_endDate=Forms! [Student_CMTs_attendance] !txt_nowDate.Ca-
ption
            And
Activity_endTime>Forms! [Student_CMTs_attendance] !txt_nowTime.Capt-
ion)
        Or
        (Activity_endDate>Forms! [Student_CMTs_attendance] !txt_nowDate.Ca-
ption))
    );

```

SQL

```

SELECT Student_lastName & " " & Student(firstName AS Name, *
FROM (Activity_record INNER JOIN Activity_information ON
Activity_record.Activity_id = Activity_information.Activity_id)
INNER JOIN Student_information ON Activity_record.student_id =
Student_information.Student_id
WHERE
(((Activity_information.Activity_startDate)=[Forms]![Student_CMTs_
attendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_startTime)<[Forms]![Student_CMTs_
attendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_endDate)=[Forms]![Student_CMTs_at-
tendancePopUp]![txt_nowDate].[Caption])) AND (((Activity_infor-
mation.Activity_endTime)>[Forms]![Student_CMTs_attend-
ancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i-
d]) And
(Activity_information.Activity_id)=[Forms]![Student_CMTs_attendan-
cePopUp]![act_id].[Caption])) AND
((Activity_record.student_id)=[Student_information].[Student_id]))

```

```

AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True)) OR
(((Activity_information.Activity_startDate)<[Forms]![Student_CMTs_
attendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_endDate)=[Forms]![Student_CMTs_at
tendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_endTime)>[Forms]![Student_CMTs_at
tendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d]) And
(Activity_information.Activity_id)=[Forms]![Student_CMTs_attendan
cePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True)) OR
(((Activity_information.Activity_startDate)=[Forms]![Student_CMTs_
attendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_startTime)<[Forms]![Student_CMTs_
attendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_endDate)>[Forms]![Student_CMTs_at
tendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d]) And
(Activity_information.Activity_id)=[Forms]![Student_CMTs_attendan
cePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True)) OR
(((Activity_information.Activity_startDate)<[Forms]![Student_CMTs_
attendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_endDate)>[Forms]![Student_CMTs_at
tendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d]) And
(Activity_information.Activity_id)=[Forms]![Student_CMTs_attendan
cePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True));

```

Student_QueryLink_CMTs_popUP

This is a linked query for showing student CMTs popup

SQL

```

SELECT *
FROM Club_schoolyear
WHERE Student_id = Forms![Student_CMTs_popUP]![txt_stdid].Caption

```

```

        AND SchoolYear =
[Forms]![Student_CMTs_popUP]![txt_schoolyear].caption
ORDER BY Club_id;

```

Student_QueryLink_CMTs_records

Student_QueryLink_CMTs_recordsPopUp_left

Student_QueryLink_CMTs_recordsPopUp_right

This is a linked query for showing student CMTs records

SQL

```

SELECT DISTINCT Activity_information.Activity_id, activity_name,
schoolYear
FROM Activity_information, Activity_record
WHERE
((Activity_information.Activity_id)=Activity_information.Activit
y_id) And
((Activity_information.Club_id)=Forms!Student_CMTs_records!txt_cl
ubid.caption))
ORDER BY SchoolYear DESC , Activity_information.Activity_id DESC;

```

SQL

```

SELECT Activity_startDate & " " & Activity_startTime AS Start,
Activity_endDate & " " & Activity_endTime AS [End],
Activity_registrationDeadline & " " &
Activity_registrationDeadlineTime AS RegDead, *
FROM Activity_information
WHERE
Activity_id=[Forms]![Student_CMTs_recordsPopUp]![txt_actid].[capt
ion];

```

SQL

```

SELECT Student_lastName & " " & Student(firstName AS Name, *
FROM Activity_record, Student_information
WHERE Activity_record.student_id = Student_information.student_id
AND
Activity_id=[Forms]![Student_CMTs_recordsPopUp]![txt_actid].[capt
ion];

```

Student_QueryLink_records_act

Student_QueryLink_records_club

This is a linked query for student records

SQL

```
SELECT *
FROM Activity_record, Activity_information
WHERE
Activity_information.Activity_id=Activity_record.Activity_id
    And
Activity_record.student_id=Forms![Student_records]!txt_stdid.Caption
    And
((Activity_endDate=Forms![Student_records]!txt_nowDate.Caption
    And
Activity_endTime<Forms![Student_records]!txt_nowTime.Caption)
    Or
(Activity_endDate<Forms![Student_records]!txt_nowDate.Caption))
    And Activity_record_attendance=TRUE
    And Activity_approval = TRUE;
```

SQL

```
SELECT *
FROM Club_record, Club_information, Club_schoolyear
WHERE Club_information.Club_id=Club_record.Club_id
    AND (Club_schoolyear.Club_id=Club_record.Club_id and
Club_schoolyear.SchoolYear=Club_record.SchoolYear)
    And Club_record.student_id=
[Forms]![Student_records]![txt_stdid].Caption
    And Club_record.SchoolYear <>
[Forms]![Student_records]![txt_schoolyear].caption
    And Club_approval = TRUE
ORDER BY Club_record.SchoolYear DESC , Club_information.Club_id;
```

Student_QueryLink_selfAccount

This is a linked query for student self account

SQL

```
SELECT Student_selfAccount
FROM Student_selfAccount
WHERE
Student_id=[Forms]![Student_Personal_selfAccount]![txt_stdid].caption;
```

Teacher_QueryLink_AMTs_activityRecords

Teacher_QueryLink_AMTs_activityRecordsPopUp_left

Teacher_QueryLink_AMTs_activityRecordsPopUp_left

This is a linked query for showing Teacher AMTs activity records

SQL

```
SELECT *
FROM Activity_information
WHERE Teacher_id =
Forms!Teacher_AMTs_activityRecords!txt_teacherid.caption
    and Activity_id NOT IN(SELECT Activity_id
FROM Activity_information
WHERE Teacher_id =
Forms!Teacher_AMTs_activityRecords!txt_teacherid.caption
    AND ((Activity_startDate =
[Forms]![Teacher_AMTs_activityRecords]![txt_nowDate].caption
        AND Activity_startTime >
[Forms]![Teacher_AMTs_activityRecords]![txt_nowTime].caption)
        OR(Activity_startDate >
[Forms]![Teacher_AMTs_activityRecords]![txt_nowDate].caption)))
ORDER BY SchoolYear DESC , Activity_information.Activity_id DESC;
```

SQL

```
SELECT Activity_startDate & " " & Activity_startTime AS Start,
Activity_endDate & " " & Activity_endTime AS [End],
Activity_registrationDeadline & " " &
Activity_registrationDeadlineTime AS RegDead, *
FROM Activity_information
WHERE
Activity_id=[Forms]![Teacher_AMTs_activityRecordsPopUp]![txt_acti
d].[caption];
```

SQL

```
SELECT Student_lastName & " " & Student(firstName AS Name, *
FROM Student_information INNER JOIN Activity_record ON
Student_information.Student_id = Activity_record.student_id
WHERE
((Activity_record.student_id)=Student_information.student_id)
And
((Activity_record.Activity_id)=Forms!Teacher_AMTs_activityRecords
PopUp!txt_actid.caption));
```

Teacher_QueryLink_AMTs_editActivity

Teacher_QueryLink_AMTs_editActivityPopUp

This is a linked query for showing Teacher AMTs edit activity

SQL

```
SELECT *
FROM Activity_information
WHERE Teacher_id =
Forms!Teacher_AMTs_editActivity!txt_teacherid.caption
    AND ((Activity_startDate =
[Forms]![Teacher_AMTs_editActivity]![txt_nowDate].caption
        AND Activity_startTime >
[Forms]![Teacher_AMTs_editActivity]![txt_nowTime].caption)
        OR (Activity_startDate >
[Forms]![Teacher_AMTs_editActivity]![txt_nowDate].caption))
ORDER BY SchoolYear DESC , Activity_information.Activity_id DESC;
```

SQL

```
SELECT *
FROM Activity_information
WHERE
Activity_id=Forms![Teacher_AMTs_editActivityPopUp]!txt_actid.Caption;
```

Teacher_QueryLink_CMTs_popUP

This is a linked query for showing Teacher CMTs Pop Up

SQL

```
SELECT *
FROM Club_schoolyear
WHERE Teacher_id = Forms![Teacher_CMTs_popUP]![txt_teaid].Caption
    AND SchoolYear =
[Forms]![Teacher_CMTs_popUP]![txt_schoolyear].caption
ORDER BY Club_id;
```

Teacher_QueryLink_CMTs_studentAttendanceOverview

Teacher_QueryLink_CMTs_studentAttendancePopUp

This is a linked query for showing Teacher CMTs student attendance

SQL

```

SELECT DISTINCT Activity_information.Activity_id, activity_name,
Activity_endDate & " " & Activity_endTime AS [End]
FROM Activity_information, Activity_record
WHERE
Activity_information.Activity_id=Activity_record.Activity_id
    AND
Activity_information.Club_id=Forms! [Teacher_CMTs_studentAttendance]!
e]!txt_clubid.Caption
    AND Activity_display=TRUE
    AND Activity_approval=TRUE
    AND (
        ((Activity_startDate=Forms! [Teacher_CMTs_studentAttendance]!txt_n
owDate.Caption
            And
Activity_startTime<Forms! [Teacher_CMTs_studentAttendance]!txt_no
wTime.Caption)
            Or
        (Activity_startDate<Forms! [Teacher_CMTs_studentAttendance]!txt_no
wDate.Caption))
            AND
        ((Activity_endDate=Forms! [Teacher_CMTs_studentAttendance]!txt_no
wDate.Caption
            And
Activity_endTime>Forms! [Teacher_CMTs_studentAttendance]!txt_noTi
me.Caption)
            Or
        (Activity_endDate>Forms! [Teacher_CMTs_studentAttendance]!txt_noD
ate.Caption))
    );

```

SQL

```

SELECT Student_lastName & " " & Student(firstName AS Name, *
FROM (Activity_record INNER JOIN Activity_information ON
Activity_record.Activity_id = Activity_information.Activity_id)
INNER JOIN Student_information ON Activity_record.student_id =
Student_information.Student_id
WHERE
(((Activity_information.Activity_startDate)=[Forms]! [Teacher_CMTs_
_studentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_startTime)<[Forms]! [Teacher_CMTs_-
studentAttendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_endDate)=[Forms]! [Teacher_CMTs_ st

```

```

udentAttendancePopUp]![txt_nowDate].[Caption]) AND
((Activity_information.Activity_endTime)>[Forms]![Teacher_CMTs_st
udentAttendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d] And
(Activity_information.Activity_id)=[Forms]![Teacher_CMTs_studentA
ttendancePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True)) OR
(((Activity_information.Activity_startDate)<[Forms]![Teacher_CMTs_
studentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_endDate)=[Forms]![Teacher_CMTs_st
udentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_endTime)>[Forms]![Teacher_CMTs_st
udentAttendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d] And
(Activity_information.Activity_id)=[Forms]![Teacher_CMTs_studentA
ttendancePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True)) OR
(((Activity_information.Activity_startDate)=[Forms]![Teacher_CMTs_
studentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_startTime)<[Forms]![Teacher_CMTs_
studentAttendancePopUp]![txt_nowTime].[Caption])) AND
((Activity_information.Activity_endDate)>[Forms]![Teacher_CMTs_st
udentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d] And
(Activity_information.Activity_id)=[Forms]![Teacher_CMTs_studentA
ttendancePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True)) OR
(((Activity_information.Activity_startDate)<[Forms]![Teacher_CMTs_
studentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_endDate)>[Forms]![Teacher_CMTs_st
udentAttendancePopUp]![txt_nowDate].[Caption])) AND
((Activity_information.Activity_id)=[Activity_record].[Activity_i
d] And
(Activity_information.Activity_id)=[Forms]![Teacher_CMTs_studentA
ttendancePopUp]![act_id].[Caption]) AND
((Activity_record.student_id)=[Student_information].[Student_id])
AND ((Activity_information.Activity_display)=True) AND
((Activity_record.Activity_approval)=True));

```

Teacher_QueryLink_CMTs_memberRecords

Teacher_QueryLink_CMTs_memberRecordsPopUp

This is a linked query for showing Teacher CMTs member records

SQL

```
SELECT Student_lastName & " " & Student(firstName AS Name, *
FROM Student_information, Club_record
WHERE Student_information.student_id=Club_record.student_id
      AND
club_id=Forms![Teacher_CMTs_memberRecords]!txt_clubid.Caption
ORDER BY SchoolYear DESC , Club_record.student_id;
```

SQL

```
SELECT Student_lastName & " " & Student(firstName AS Name,
Student_grade & Student_class AS Class, *
FROM Student_information, Student_grade
WHERE Student_information.student_id=Student_grade.student_id
      AND
schoolYear=Forms![Teacher_CMTs_memberRecordsPopUp]!txt_schoolyear
.CCaption
      AND
Student_information.student_id=Forms![Teacher_CMTs_memberRecordsP
opUp]!txt_stdid.Caption;
```

(c) Linked Insert Queries

The following are linked insert queries.

Student_InsertRecords_Activity_information

Teacher_InsertRecords_Activity_information

This is a linked query for student and teacher insert activity information

SQL

```
INSERT INTO Activity_information ( Activity_id, Club_id,
Activity_name, Activity_type, Activity_description,
Activity_startDate, Activity_startTime, Activity_endDate,
Activity_endTime, Activity_registrationDeadline,
Activity_registrationDeadlineTime, Activity_maxParticipants,
Teacher_id, Activity_pending, Activity_location, SchoolYear )
VALUES ([Forms]![Student_CMTs_addActivity]![act_id].value,
[Forms]![Student_CMTs_addActivity]![txt_clubid].caption,
[Forms]![Student_CMTs_addActivity]![act_name].value,
[Forms]![Student_CMTs_addActivity]![act_type].value,
[Forms]![Student_CMTs_addActivity]![act_dec].value,
[Forms]![Student_CMTs_addActivity]![act_startdate].value,
[Forms]![Student_CMTs_addActivity]![act_starttime].value,
[Forms]![Student_CMTs_addActivity]![act_enddate].value,
[Forms]![Student_CMTs_addActivity]![act_endtime].value,
[Forms]![Student_CMTs_addActivity]![act_regdeaddate].value,
[Forms]![Student_CMTs_addActivity]![act_regdeadtime].value,
[Forms]![Student_CMTs_addActivity]![act_maxpart].value,
[Forms]![Student_CMTs_addActivity]![txt_teacher].caption,
[Forms]![Student_CMTs_addActivity]![act_ele].value,
[Forms]![Student_CMTs_addActivity]![act_loc].value,
[Forms]![Student_CMTs_addActivity]![txt_schoolyear].caption);
```

SQL

```
INSERT INTO Activity_information ( Activity_id, Club_id,
Activity_name, Activity_type, Activity_description,
Activity_startDate, Activity_startTime, Activity_endDate,
Activity_endTime, Activity_registrationDeadline,
Activity_registrationDeadlineTime, Activity_maxParticipants,
Teacher_id, Activity_pending, Activity_location, SchoolYear )
VALUES ([Forms]![Teacher_AMTs_addActivity]![act_id].value,
[Forms]![Teacher_AMTs_addActivity]![txt_clubid].caption,
[Forms]![Teacher_AMTs_addActivity]![act_name].value,
[Forms]![Teacher_AMTs_addActivity]![act_type].value,
[Forms]![Teacher_AMTs_addActivity]![act_dec].value,
[Forms]![Teacher_AMTs_addActivity]![act_startdate].value,
[Forms]![Teacher_AMTs_addActivity]![act_starttime].value,
[Forms]![Teacher_AMTs_addActivity]![act_enddate].value,
```

```
[Forms]![Teacher_AMTs_addActivity]![act_endtime].value,  
[Forms]![Teacher_AMTs_addActivity]![act_regdeaddate].value,  
[Forms]![Teacher_AMTs_addActivity]![act_regdeadtime].value,  
[Forms]![Teacher_AMTs_addActivity]![act_maxpart].value,  
[Forms]![Teacher_AMTs_addActivity]![txt_teacher].value,  
[Forms]![Teacher_AMTs_addActivity]![act_ele].value,  
[Forms]![Teacher_AMTs_addActivity]![act_loc].value,  
[Forms]![Teacher_AMTs_addActivity]![txt_schoolyear].caption);
```

Student_InsertRecords_Activity_record

Student_InsertRecords_Club_record

This is a linked query for student insert activity and club record

SQL

```
INSERT INTO Activity_record (Activity_Record_addedDate,  
Activity_id, Activity_approval, student_id)  
VALUES  
([Forms]![Student_browseActivitiesPopUp]![txt_nowDate].caption,  
[Forms]![Student_browseActivitiesPopUp]![act_id].caption,  
[Forms]![Student_browseActivitiesPopUp]![act_checkTpen].value,  
[Forms]![Student_browseActivitiesPopUp]![txt_stdid].caption);
```

SQL

```
INSERT INTO Club_record (Club_Record_addedDate, club_id,  
Club_approval, schoolYear, student_id)  
VALUES ([Forms]![Student_browseClubsPopUp]![txt_nowDate].caption,  
[Forms]![Student_browseClubsPopUp]![club_id].caption, TRUE,  
[Forms]![Student_browseClubsPopUp]![txt_schoolyear].caption,  
[Forms]![Student_browseClubsPopUp]![txt_stdid].caption);
```

(d) Normal Insert Queries

The following are Normal Insert Queries

InsertRecords_Activity_information

InsertRecords_Club_information

InsertRecords_Student_information

InsertRecords_Teacher_information

These are linked queries to add new information in activity, club, student, teacher

SQL

```
INSERT INTO Activity_information ( Activity_id,
club_id,
Activity_name,
Activity_type,
Activity_description,
Activity_startDate,
Activity_startTime,
Activity_endDate,
Activity_endTime,
Activity_registrationDeadline,
Activity_registrationDeadlineTime,
Activity_maxParticipants,
Teacher_id,
Activity_display,
Activity_pending,
Activity_location,
SchoolYear )
VALUES ([Input Activity ID, please insert valid value],
[Input Club ID, "C000"],
[Input Name], [Input Type],
[Input Description],
[Input Start Date, "DD/MMM/YYYY"],
[Input Start Time, "HH/MM/SS"],
[Input End Date, "DD/MMM/YYYY"],
[Input End Time, "HH/MM/SS"],
[Input Registration Deadline Date, "DD/MMM/YYYY"],
[Input Registration Deadline Time, "HH/MM/SS"],
[Input Max Participants],
[Input Teacher ID, "T000"],
[Input Description],
[Input Election],
[Input Location],
[Input School Year, "DD/MMM/YYYY"]);
```

SQL

```
INSERT INTO club_information ( Club_id,
Club_name,
Club_descriotion,
Club_pending )
VALUES ([Input Club ID, "C000", please insert valid value],
[Input Club Name],
[Input Description],
[Input Election Value, 'TRUE/False']);
```

SQL

```
INSERT INTO student_information ( student_id,
Student(firstName,
Student.lastName,
student_gender,
Student_email,
Student_dob,
Student_phoneNumber )
VALUES ([Input Activity ID, please insert valid value],
[Input First Name],
[Input Last Name],
[Input Gender, 'M'/ 'F'],
[Input Email],
[Input Date of Birth, "DD/MMM/YYYY"],
[Input Phone Number]);
```

SQL

```
INSERT INTO Teacher_information ( Teacher_id,
Teacher_lastName,
Teacher(firstName,
Teacher_email,
Teacher_gender,
Teacher_phoneNumber )
VALUES ([Input Teacher ID, "T001", please insert valid value],
[Input Last Name ],
[Input First Name],
[Input email],
[Input Gender, 'M/ F'],
[Input Phone Number]);
```

8. Implementation Details

Software used: Microsoft Access, figma, Discord

What software do we use?	Why do we use that software?
Microsoft Access	To create the whole system
Figma	To create buttons and background
Discord	To get in voice call anytime and chat with groupmate on how to create the database system. And with the share screen function, group mates who are more familiar with the database can show other group mates how to do it by displaying how they did certain parts.
Excel	Easier to process raw data, therefore, we use it to process raw data first. After that, we will import it to our Access database file.

Source of data: Generate possible data for system test and unit test by ourselves

We sometimes got to Microsoft support ,overseas forums like stack overflow and tutorials from Youtube to find certain ways to do particular functions.

What software do we use?	How to install
Figma	It can be used in the web browsers like Firefox and Google Chrome, but you need to use a Google account to login to Figma in order to create pictures.
Discord	Go to https://discord.com/download And then download discord and create an account

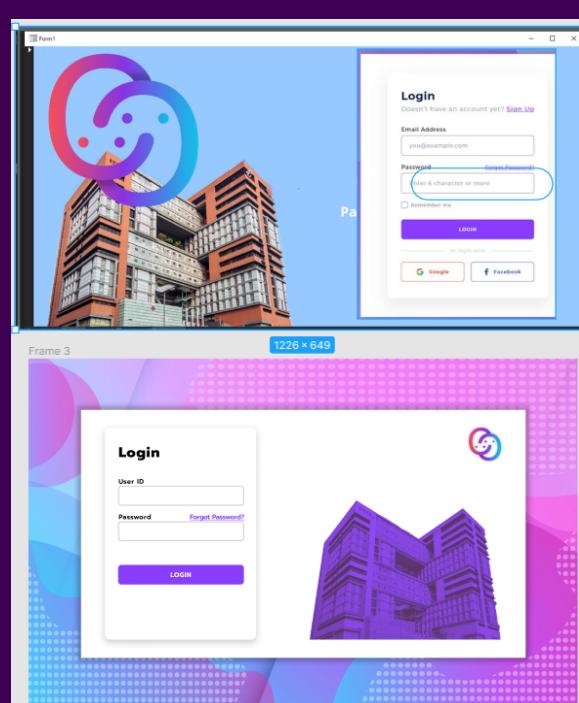
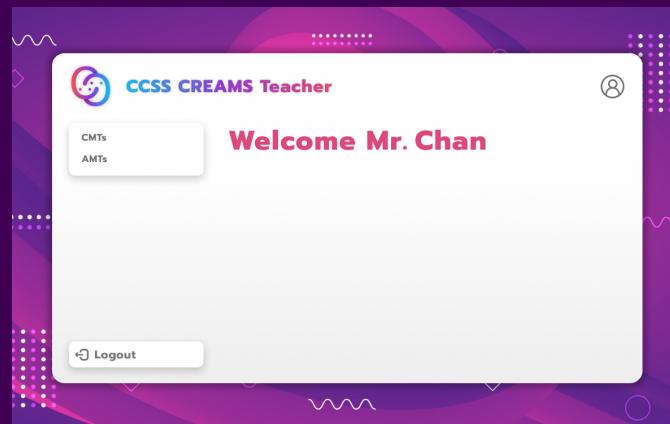
9. Difficulties

(a) Design phase

Design Teacher Features:

We only have students' experience. For the administration of the school, we have no idea.

We tried to connect to our secondary school teachers. As they are super busy, they cannot have a short meeting with us, so it further adds the difficulties of implementing how teachers use this system and how to make their life more convenient with features. Therefore, we need to dig deeper on the Internet to figure out school operations and design a usable application for teachers.



Design backgrounds and buttons:

This process has a lot of design to do, and it takes many times as we finish the background design. As we want to make a custom system, we need to use a new tool, Figma.

Figma is a collaborative web-based vector editor and UX design tool.
(<https://en.wikipedia.org/wiki/Figma>)

We must ask a couple of our friends to comment on those pictures if they look fascinating and eye-catching or not.

After that, we need to modify those pictures.

These processes run in a circle and we modify them so many times.

After that, we need to ensure that each button is an individual picture file and import into the database.

We need to map those buttons tidily first.

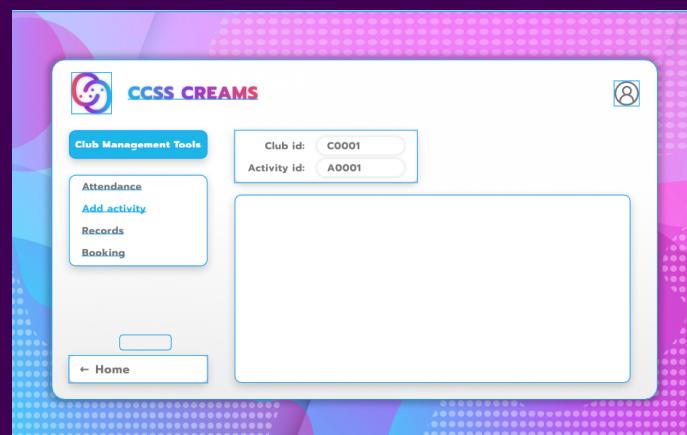
Finally, add events to each button range so that when the user clicks the button, it will react with the suitable process.

User friendly UI:

After knowing the school's operation, we need to create a user-friendly UI to let those workers that are not familiar with the internet use it.

We need to make sure everybody, especially someone who is not familiar with using computers and systems, can also use it.

The hard part of this is that sometimes we assume people can use this as we think everybody is familiar with this system. We have to think twice and draw out what users will do in this process and react. This is not easy because we do not have users who do not implement this database to test the system. That means the user acceptance test cannot be done just by our group, so we have to try everything to ensure there is enough guidance for users in the systems.



VBA design:

First, since no one in our group has learned the VBA before, it took a lot of time to learn it from overseas forums and Youtube and try those in our system's function.

Especially in overseas forums, it is hard to get what we want because many more skilled people have already solved similar problems. There are often several ways to do it and we need to compare which works the most in our situation.

As shown in this picture

```
Option Compare Database
Option Explicit

Public TargetForm As String
Public CurrentId As String
Public CurrentTMPid As String

Public Sub setTargetForm(tmp As String)
    TargetForm = tmp
End Sub

Public Function getTargetForm() As String
    getTargetForm = TargetForm
End Function

Public Function ToTempPage(tmp As String)

    DoCmd.Close
    TargetForm = tmp
    DoCmd.OpenForm "Tranto_tempPage"
End Function

Public Function getNowDate() As String
    getNowDate = Format(Now, "dd") & "/" & Format(Now, "mm") & "/" & Format(Now, "yyyy")
End Function
```

VBA is very crowded and not like other languages like C++, there are only two types of colors and it is tough to watch and very tiring to watch for a long time.

With almost only black words, it is hard to know whether we type the correct function like DoCmd and some other parts.

Also, VBA does not like that the other language can debug it first to check syntax errors. It is hard to know unless you run the form one time.

Sometimes, when designing the VBA, we need to check if it worked after a specific amount of function is added. It takes a lot of time to run it once again to prevent any syntax errors that blocked the system from working right away.

(b) Implementation Phase**No related experience:**

Since none of our groupmates have taken the database as an elective in ICT in secondary school, we cannot do some more complicated questions just by ourselves with only limited time in lectures and tutorials. So we spent a lot of time watching videos, articles, and forums to catch up and find what we wanted and implement it to the database.

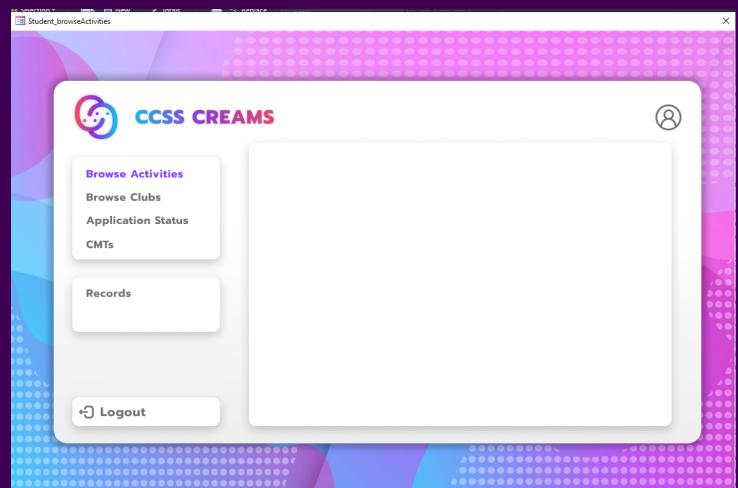
Debug:

A functional system should not have any errors, so we have to guarantee everything is perfect. We need to avoid all the mistakes, which took up so much time. Other than syntax errors, there are errors that the system performed differently than what we've expected. So it takes a long time to try different ways to solve the problem.

Bug:

If we open a form related to the last action query, a form will show no information. Only this bug haunted us for a couple of hours. We tried to go to the home page automatically and go back to the current form, but the information still cannot be loaded.

After that, we tried to refresh the form and refresh the query, but that information still cannot be loaded. Finally, we found out that the best way to solve this problem is to create a transition form to replace the last action.



Expectation:

Creating a real-life database system is more complex than our imagination. First, It is impossible to build an ideal system as there are so many possible functions added in this database, so we guess that's why many companies still have so many updates even though the internet service is already out. And it is hard to put a line on where to stop adding functions as we want this system to be as good as possible.

Secondly, The relationship among the tables and SQL statements is complex, requiring many times to be familiar with real-life situations. Because with an actual usable system, there should not be just 5-6 tables with just a couple of queries like what we did in lesson exercises, tests, and exams.

Thirdly, to make a custom database system, we have to learn how to use new tools and new language to make the system more user-friendly. We have learned new things like using Figma to create backgrounds and buttons and using VBA to let the system react when users click those buttons.

(c) Lesson learnt**Division of labour:**

After knowing the system's hardness, we decide to divide the work into multiple parts to reduce each groupmate's workload. We face different problems, such as not following the schedule. Each groupmate's ability may not be the same, so that a good division of labour is essential, which can maximize the progression.

Used a microsoft planner to plan each part's deadlines to add momentum to increasing the work speed. We tried to talk with each other by meeting each deadline through voice chat in discord with each other.

Testing:

Thirdly, at first, we think testing is just like correcting a wrong math question. We need to create a system into different parts and do unit tests to remove logic errors and syntax errors. We need to test the integration of those parts as a system to check if there are any errors between each unit. Unfortunately, we cannot get a user to do a user acceptance test, as we mentioned before. We found out that it takes a lot of time, even though we created all the tables, queries, and VBA functions.

10. Work distribution list

Task Name	Start Date	Finish Date	Responsible
Data Generate	14/3	16/3	Stanley, Richard
Access Query	18/3	5/4	Ricky, Richard, Stanley, Jesse
Access Table	11/3	13/3	Jesse, Stanley
Access Form	18/3	7/4	Jesse, Ricky
Access Report	9/4	10/4	Richard, Ricky
Design UI	10/3	14/3	Jesse, Richard
Revise	11/4	12/4	Stanley, Ricky
Testing	13/4	14/4	Ricky, Richard
Report	11/4	15/4	Richard, Jesse
PowerPoint	10/4	15/4	Stanley, Ricky

11. Conclusion

We have a lot of business rules setted at the beginning of the project, we thought this is not possible to fulfil. We encountered a lot of issues through the project and we tackled them one by one. We keep refining and improving the project. With the help of a time table and Microsoft Planner, this project meets all the targets flawlessly at the end. We finally are comfortable to submit this project.

We have learnt so many from doing this project. **Planning** a project, **designing** a database and graphics, and most importantly -- **communicating** with groupmates. We are happy that we have this opportunity to do this project

We definitely although we just learnt the Access and SQL in the lectures and tutorials but creating a database needs more than that. As we mentioned before we use figma to create visual design, we use VBA to make every click react the way that we wanted and everyone for comments using discord. That's why we say that we've learnt a lot even outside of the database. For example, the design touch of making an eye-catching system display even changes the fonts to fit the display. We also need to know how to quickly communicate with each other. Sometimes we need to command each other to make our work flow be quicker. This is something that we cannot learn just attending lessons. And we are happy about what we've learnt.

Finally, we are satisfied with finishing this project because we've done our best to make this group project as ideal as possible within our ability and the limitation of time. We know this system is not perfect and it needs a lot of improvement so We think after handing in this project, we will keep improving the system and hope it can be better than before.

In the Future:

We can add a payment system that student can pay the fee

We can also create a room booking system for booking venue for activities

We can add school officers for collecting fees and modify certain important activities that teacher cannot be modified

We also can create a calendar for students so that they know when to go to the activity they signed up for.

12. Reference

Name List CC Moodle Page: <https://moodle.cpce-polyu.edu.hk/user/index.php?id=141907>

Graphics and image:

<https://www.vecteezy.com/vector-art/682270-pink-purple-geometric-gradient-fluid-style-wallpaper>

<https://www.vecteezy.com/vector-art/695806-purple-and-pink-abstract-fluid-gradient-background>

https://www.freepik.com/free-vector/fluid-gradient-purple-shape-wallpaper-background-design_8038883.htm

<https://wallpaperaccess.com/abstract-triangle>