WORKSHOP 1

Company Event and Training Registration Management System

LOW XIAO QIAN -B032210182

BITM S1G1

Chapter 1

INTRODUCTION

Project Background

The Company events and training registration management system is designed to manage the registration process of various types of company events such as internal training, seminars, and workshop. This system provide efficiency to collect events details, tutors' information and staff information. The purpose of creating a system for registration is to build a platform for people easy to search and apply events.

Project Statement

Many organizers still rely on manual methods such as using paper-based registrations and email, to distribute event information. This may lead to mistakes and inefficiencies because the time required to distribute event information is longer. And mistakes are not allowed since it may result in confusion among tutors and staffs and will even affect the event's reputation.

Objectives

- To study the problem that is faced by the company for managing the registration of educational events and training.
- To develop and design company events and training registration management system.
- I.To generate report for company events and training registration management system that was created for this project.

Project Significance

- Reducing the risk of miscommunication
- Improving event management efficiency
- Reach company's strategic goals

Target users

Admin

Tutor

Staff

Scopes

Sign up

Admin, tutors and staff can register their own information by using user ID, username, age, gender, phone number and password.

Login

Admin, tutors and staff can login by entering the correct username and password.

Add/ Delete/ Update

Admin able to add, delete, and update tutor, staff and events information. Tutors can add and update events and their own information in the system, while staff can only update their own information.

Display table

Admin able to display all the records in the system. Tutors can view their events, tutors list, staff list and list of attendees that registered for their event. staffs are only able to display the events list to registration.

Chapter 2

SYSTEM ANALYSIS

Problem Description

As we know there are many companies still using manual methods to record the registration of the events such as seminars, workshop and internal training. A lot of time will be spent on managing the process of registration. Some mistakes might happen when using manual method, such as missing documents, since they mostly use paperwork to record and cause inefficiency.

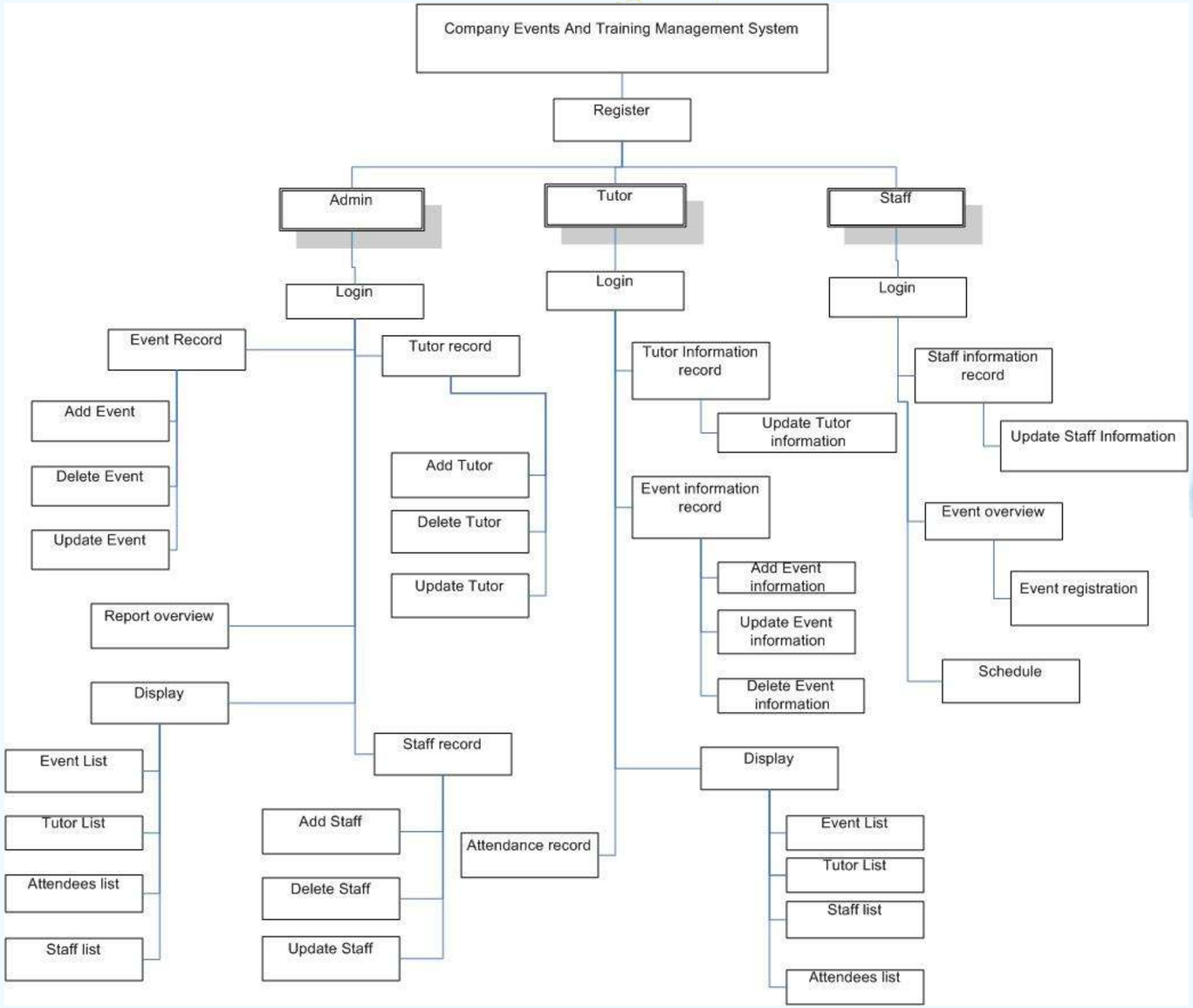
Problem Decomposition

- Admin able to manage all information of admin, tutor, staffs and events by add, delete and update.
- Tutor and staff can only manage their own information.
- Admin and tutor able to view total number of staffs who attend the events that provide by company easily that have record by system.

Project Significance

1. Reducing the risk of miscommunication
2. Improving event management efficiency
3. Reach company’s strategic goals

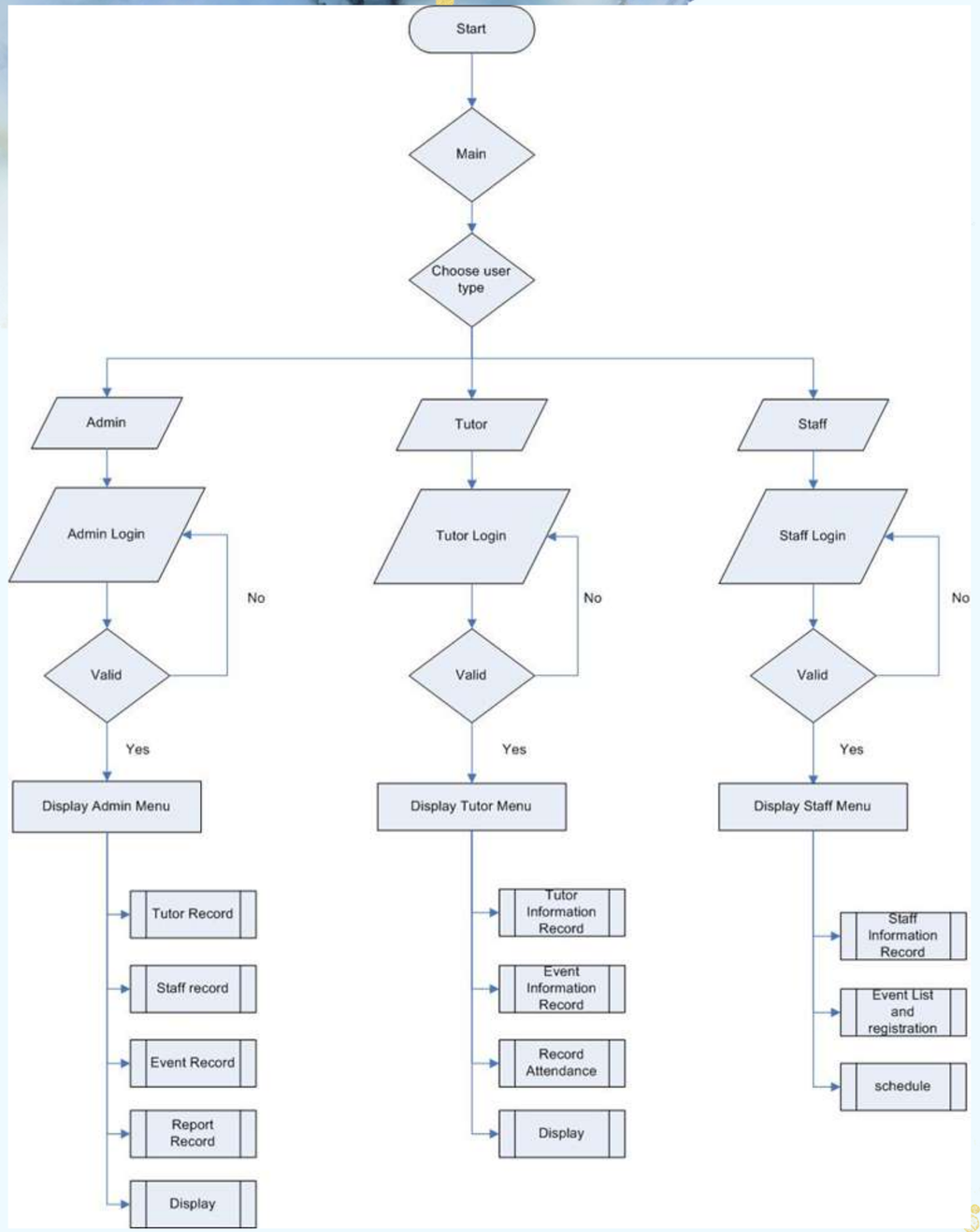
Structure Chart



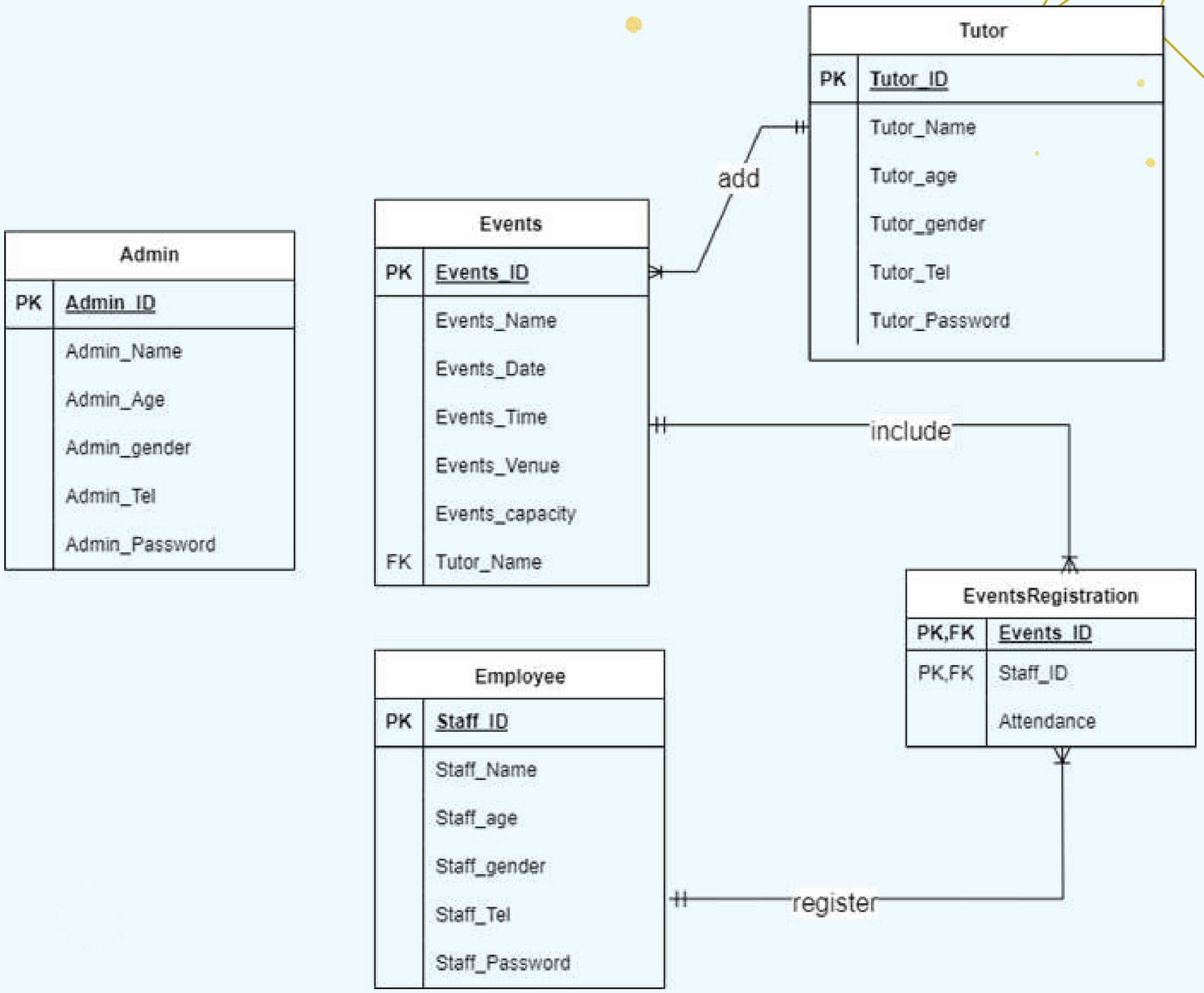
Chapter 3

System Design

Flow Chart


Start
Main
Choose user type
Admin
Tutor
Staff
Admin Login
Tutor Login
Staff Login
No
No
No
Valid
Valid
Valid
Yes
Yes
Yes
Display Admin Menu
Display Tutor Menu
Display Staff Menu
Tutor Record
Staff record
Event Record
Report Record
Display
Tutor Information Record
Event Information Record
Record Attendance
Display
Staff Information Record
Event List and registration
schedule

Entity Relation Diagram (ERD)



Data Dictionary

Admin

Attribute name	Data type and size	Range	Not null?	Default value	Unique?	PK or FK	FK reference table
Admin_ID	INT (11)		Yes		Yes	PK	
Admin_Name	VARCHAR2 (50)		Yes				
Admin_Age	INT(3)		Yes				
Admin_Gender	VARCHAR2 (7)		Yes				
Admin_Tel	VARCHAR2 (20)		Yes				
Admin_Password	VARCHAR2 (20)		Yes		Yes		

Tutor

Attribute name	Data type and size	Range	Not null?	Default value	Unique?	PK or FK	FK reference table
Tutor_ID	INT (11)				Yes	PK	
Tutor_Name	VARCHAR2 (50)		Yes				
Tutor_Age	INT(3)		Yes				
Tutor_Gender	VARCHAR2 (7)		Yes				
Tutor_Tel	VARCHAR2 (20)		Yes				
Tutor_Password	VARCHAR2 (20)		Yes		Yes		

Events

Attribute name	Data type and size	Range	Not null?	Default value	Unique?	PK or FK	FK reference table
Events_ID	INT(11)				Yes	PK	
Events_Name	VARCHAR2 (50)		Yes				
Events_Date	VARCHAR2 (20)		Yes				
Events_Time	VARCHAR2 (20)		Yes				
Events_Venue	VARCHAR2 (100)		Yes				
Events_capacity	INT (11)		Yes				
Tutor_ID	INT(11)		Yes			FK	Tutor

Staff

Attribute name	Data type and size	Range	Not null ?	Default value	Unique ?	PK or FK	FK reference table
Staff_ID	INT (11)				Yes	PK	
Staff_Name	VARCHAR 2 (50)		Yes				
Staff_Age	INT (3)		Yes				
Staff_Gender	VARCHAR 2 (7)		Yes				
Staff_Tel	VARCHAR 2 (20)		Yes				
Staff_Password	VARCHAR 2 (20)		Yes		Yes		

EventsRegistration

Attribute name	Data type and size	Range	Not null?	Default value	Unique?	PK or FK	FK reference table
Events_ID	INT(11)					PK,FK	Events
Staff_ID	INT(11)					PK,FK	Staff
Attendance	INT(3)		Yes				

Chapter 4

System Implementation

Selection

```
MainMenu()

system("cls");
char user;
cout << "\t\t=============================================================================================" << endl;
cout << "\t\t|  Welcome to Company Events and Training Registration Management System                    |" << endl;
cout << "\t\t=============================================================================================" << endl;
cout << "\t\t|1. Admin                                                                                   |" << endl;
cout << "\t\t|2. Tutor                                                                                   |" << endl;
cout << "\t\t|3. Staff                                                                                   |" << endl;
cout << "\t\t|4. Exit                                                                                    |" << endl;
cout << "\t\t=============================================================================================" << endl <<
cout << "\t\tChoice: ";
cin >> user;

switch (user)
{
case '1':
    AdminMainMenu();
    break;
case '2':
    TutorMainMenu();
    break;
case '3':
    StaffMainMenu();
    break;
case '4':
    cout << "Exiting..." << endl;
    exit(0);
default:
    cout << endl << "Please enter a number between 1 - 4 to continue."<<endl;
    system("pause");
    system("cls");
```

Array

```
lass::Register(string user)

alphaCount=0, numCount=0;
ng checkkey;
em("cls");
ignore();
 << "\t\t=============================================================================================" << endl;
 << "\t\t                                Welcome " + user + "                                         " << endl;
 << "\t\t=============================================================================================" << endl;
 << "\t\tPlease insert following information to register: " << endl;
 << "\t\t1.Name          : ";
ine(cin, Name);

 << "\t\t2.Age           : ";
ine(cin, age);

 << "\t\t3.Gender        : ";
ine(cin, gender);

 << "\t\t4.Contact numbers : ";
ine(cin, Tel);
 << "\t\t5.Password (Your password should contain letter and number) : ";
ine(cin, Password);
t char* passwordArray = Password.c_str();
(int i = 0; i < Password.length(); i++)

if (isalpha(passwordArray[i]))
{
    ++alphaCount;
}
else if(isdigit(passwordArray[i])) {
    ++numCount;
}

e (alphaCount >= 1 && numCount >= 1) {
```

Pointer

```
public:
    int userID;
    int qstate = 0;
    MYSQL* conn = nullptr;
    MYSQL_ROW row = nullptr;
    MYSQL_RES* res = nullptr;
```

Function

```
void AdminMainMenu();
void TutorMainMenu();
void StaffMainMenu();
void MainMenu();
void AdminMenu();
void StaffMenu();
void record(string);
void userMenu(string);
void TutorMenu();
void tutorrecord();
void Staffrecord();
void eventrecord(string);
void display(string);
void eventdisplay();
void report();
```

Calculation

```
else if (pilihan == 2)
{
    ofstream file("Monthly_percentage_of_attendance.txt");
    int month = 1;

    file << "-------------------------------------------------------------------------------------------------------------------------------------------------------" << endl;
    file << "                                                Monthly percentage of attendance                                                                       " << endl;
    file << "-------------------------------------------------------------------------------------------------------------------------------------------------------" << endl;

    file << setw(10) << "Month" << setw(25) << "Total organized events" << setw(25) << "Overall capacity" << setw(26) << "Numbers of attendees" << setw(25) << "Percentage of attendance" << endl;
    do
    {
        double TotalCapacity = C.Ttotal_capacity(to_string(month));
        double Totalevent = C.total_eventPM(to_string(month));
        double TotalAttend = C.total_attend(to_string(month));
        double ave = (TotalAttend / TotalCapacity) * 100;
        if (isnan(ave)) {
            ave = 0;
        }
        ostringstream tempStream;
        tempStream << fixed << setprecision(2) << ave;
        file << setw(5) << month << setw(25) << Totalevent << setw(25) << TotalCapacity << setw(25) << TotalAttend << setw(20) << tempStream.str() << endl;
        month++;
    } while (month != 13);
    cout << "Report generated successfully." << endl;
    system("pause");
    file.close();
    report();
}
```

control

```
if (T_record == 1)
{
    cin.ignore(256, '\n');
    char add = C.add(user);
    while (add == 'y' || add == 'Y')
    {
        add = C.add(user);

    }
    if (add == 'n' || add == 'N')
    {
        system("cls");
        cout << "\n\tPress enter to back to " + user + " record menu....." << endl;
        cout << "=================================================";
        cin.get();
        record(user);
    }
}
else if (T_record == 2)
{
    char cont;
    do {
        char back = C.deleteData(user);
        if (back == 'Q' || back == 'q')
        {
            record(user);
        }
        else if (back == 'T' || back == 't')
        {
            back = C.deleteData(user);
        }

        cout << "\n\nDo you want to delete other " + user + "? (Y/N):";
        cin >> cont;
    } while (cont == 'y' || cont == 'Y');
    system("cls");
    cout << "\n\tPress enter to back to " + user + " record menu....." << endl;
    cout << "=================================================";
    cin.get();
    record(user);
}
```

Error Handling

```
if (!C.qstate)
{
    cout << "=============================================================================================================================" << endl;
    cout << "                                                Event List                                                                   " << endl;
    cout << "=============================================================================================================================" << endl;
    cout  << setw(11) << "Event ID" << setw(30) << "Name" << setw(15) << "Date" <<setw(20)<<"Time" << setw(30) << "Venue" << setw(15) << "Capacity" << setw(30) << "Tutor name" << endl;
    C.res = mysql_store_result(C.conn);
    while (C.row = mysql_fetch_row(C.res))
    {
        cout  << setw(11) << C.row[0] << setw(30) << C.row[1] << setw(15) << C.row[2] <<setw(20)<<C.row[3] << setw(30) << C.row[4] << setw(15) << C.row[5] << setw(25) << C.row[6] << endl;
    }
}
cout << "\nEnter event ID to register or '0' to back to Employee Menu:";
cin >> reg;

if (reg == 0)
{
    EmployeeMenu();
}
else {
    string check_query = "SELECT * FROM eventsregistration WHERE events_ID = " + to_string(reg) + " AND Employee_ID = " + to_string(C.userID);
    const char* check_query_cstr = check_query.c_str();
    C.qstate = mysql_query(C.conn, check_query_cstr);
    if (C.qstate == 0)
    {
        C.res = mysql_store_result(C.conn);
        if (C.res != nullptr && mysql_num_rows(C.res) > 0)
        {
            cin.ignore();
            cout << "\n\nYou are already registered for this event. Please choose another event.\n";
            cout << "\nPress enter to select another event." << endl;
            cin.get();
            eventdisplay();
            return;
        }
    }
    else
    {
```

CONCLUSION

In conclusion, the Company Events Registration Management System is a user-friendly system that is programmed for people to simplify managing the process of registering company events. This system can reduce time taken on collecting data of events and information of staffs. Hence, Admin and tutor can oversee details of company events and list of staffs easily to make an accurate arrangement.

DEMONSTRATION

THANK YOU