**Python Database Integration Project**

**Event Management System**

**Team size: 3 Max Mark: 50**

**Instructions to Candidates:**

1. Follow the class diagram strictly. Read the problem statement, functionality and the other details provided carefully and implement the solution
2. Ensure that all the constructors are public and DO NOT change the prototype of the constructor and the methods
3. Code submitted with compilation errors may not get evaluated
4. The coding standards are mandatory wherever applicable

**Guidelines:**

1. This assessment is intended to test the hands on skills related to Programming and Database Connectivity using Python
2. You must use Eclipse, cx\_Oracle and Oracle 11g Express Edition for implementing

**NOTE: NOT adhering to the above instructions and guidelines may lead to drastic reduction in your score even if the code is executing without any errors**

**Few errors which might result in mark reduction:**

1. Infinite loops/infinite recursive calls in the code
2. Class diagram violation
3. Improper constructor prototype or definition

***WISH YOU ALL THE BEST***

**Problem Description:**

**Event Management system:**

Event Management Applicaiton  keeps track of necessary information required for an event.

Information includes event details, organizer details, volunteer details, visitor details.

Event details contains EventName, Date, Duration, Place, Website.

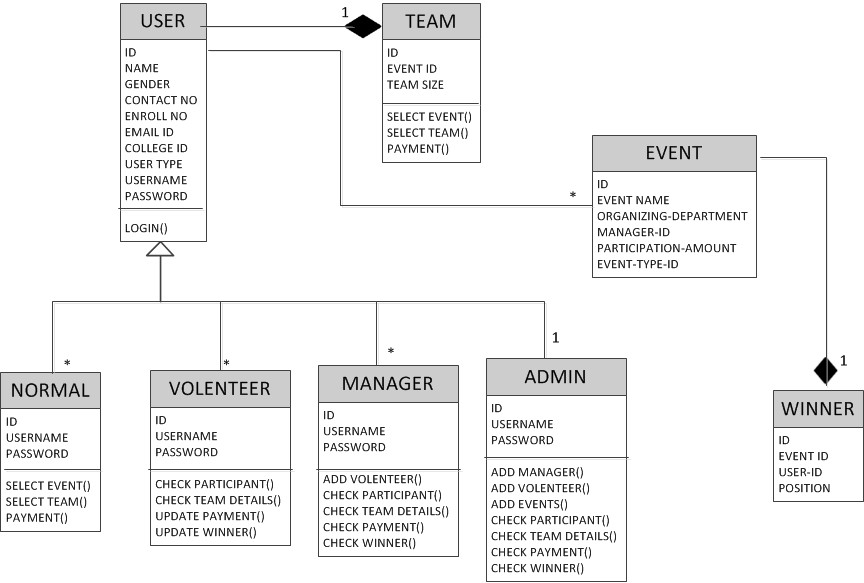
Organizer details contains EventName, OrganizerName, Address, ContactNo, EmailAddress, Role.

Volunteer details contians EventName, VolunteerName, Address, ContactNo, EmailAddress, Role.

Visitor details contains EventName, VisitorName, Address, ContactNo, EmailAddress, Role.

Write a Python program to implement the class diagrams below.

**Class Diagram**

****

**Database:** Implement the above class diagram and connect it with database tables to maintain event details. Validate the event details and. Demo (Main class) must have the option to select for manipulating with database like insert/modify/delete/search/report.

**Table Structure:**

Steps to be followed in MySQLdb

1. Create a new database with name ‘management’

mysql> create database management;

2. Change to management database

mysql> use management;  
Database changed

3. Create four tables in management database with the following names

i.  event

ii. organizer

iii. volunteer

iv. visitor

4. Event table should contain the following attributes with one primary key

a. EventName

b. Date

c. Duration

d. Place

e. Website

Primary Key – EventName

create table event (  
EventName varchar(30) NOT NULL,  
Date date NOT NULL,  
Duration time NOT NULL,  
Place varchar(30) NOT NULL,  
Website varchar(30) NOT NULL,  
PRIMARY KEY  (EventName));

5. Organizer table should contain the following attributes with one foreign key

a. EventName

b. OrganizerName

c. Address

d. ContactNo

e. EmailAddress

f. Role

Foreign Key – EventName

create table organizer (  
EventName varchar(30) NOT NULL,  
OrganizerName varchar(30) NOT NULL,  
Address varchar(30) NOT NULL,  
ContactNo int(11) NOT NULL,  
EmailAddress varchar(30) NOT NULL,  
Role varchar(30) NOT NULL,  
FOREIGNKEY(EventName) REFERENCES event (EventName));

6. Volunteer table should contain the following attributes with one foreign key

a. EventName

b. VolunteerName

c. Address

d. ContactNo

e. EmailAddress

f. Role

Foreign Key – EventName

create table volunteer (  
EventName varchar(30) NOT NULL,  
VolunteerName varchar(30) NOT NULL,  
Address varchar(30) NOT NULL,  
ContactNo int(11) NOT NULL,  
EmailAddress varchar(30) NOT NULL,  
Role varchar(30) NOT NULL,  
FOREIGNKEY(EventName) REFERENCES event (EventName));

7. Visitor table should contain the following attributes

a. EventName

b. VisitorName

c. Address

d. ContactNo

e. EmailAddress

f. Role

Foreign Key – EventName

create table visitor (  
EventName varchar(30) NOT NULL,  
VolunteerName varchar(30) NOT NULL,  
Address varchar(30) NOT NULL,  
ContactNo int(11) NOT NULL,  
EmailAddress varchar(30) NOT NULL,  
Role varchar(30) NOT NULL,  
FOREIGNKEY(EventName) REFERENCES event (EventName));

**Note:**

1. Use **case insensitive String** comparison wherever applicable.
2. Perform **case sensitive character** comparison wherever applicable.
3. Do not include any extra instance variables or member methods in the given classes.
4. Do typecasting wherever appropriate
5. The order of passing the values to the child class constructor would be member variables of parent class followed by the child class
6. **Implement Getter and Setter methods for all the variables**

**Implementation Details:**

**Class: event**

Add code to event based on the class diagram. Wherever needed, the additional implementation details are given below.

* Add eventname of new event to be organized and date,duration,place, website of that particular event should be included in this class

**Class: organizer**

**Organizer name()**

* This method takes organizer name and all the details of the organizer to the database using variables

**Class: Volunteer**

* This method takes volunteer for each event with volunteer name as object. This also includes the details of the volunteer.

**Demo Class:**

* Demo class is the starter class
* Code for the Demo class is provided to you
* Read and understand the functionality of the code
* You can modify the code of Demo class for testing purpose but ensure that you are submitting it compilation error free
* Demo class will not be evaluated
* Demo class will have the option to insert/modify/delete/search customer details

**Sample Output Expected:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Event Management System**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. User Login
2. Event creation
3. Volunteer Allocation
4. Cancel Event
5. Report

&=&=&=&=&=&=&

**Marks Distribution [50 Marks]:**

1. Implementation of Customer class : **10 Marks**
   1. Declaration of instance variables : 4 mark
   2. Getter & Setter methods : 2 mark
   3. Validate methods : 4 marks
2. Table Creation in Database : **10 Marks**
3. Implementation of Staff class : **10 Marks**
   1. Declaration of instance variables : 4 marks
   2. Getter & Setter methods :2 mark
   3. Validate methods :4 marks
4. Implementation of **event** class : **10 Marks**
5. Volunteer table : 2 marks
6. Organizer table : 2 marks
7. Cancel event : 2 marks
8. Report for each event : 2 marks
9. Coding Standard and Naming Convention **: 10 Marks**