AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY
School of Engineering

ADVANCED PROGRAMMING (19CSE201)

DBMS (19CYS204)

MOVIE TICKET BOOKING

Prepared by

TEAM 11

R.UDAYKIRAN(CH.EN.U4CYS208)
V.SOHITH(CH.EN.U4CYS20082)
REDDYMALLA.GYANENDHAR(CH.EN.U4CYS20064)
YESWANTH.UPPUTURI(CH.EN.U4CYS20086)

Under the Guidance of

Dr. Ragupathy P
Dr.Vigneshwaran Muralidaran

Assistant Professors
Department ofComputer Science
& Engineering

अद्धावान् लभते ज्ञानम्
AMRITA VISHWA VIDYAPEETHAM
UNIVERSITY
Established u/s 3 of UGC Act 1956

ACKNOWLEDGEMENT

We group 11 would like to express our special thanks of gratitude to our professors Dr. Raupathy sir and Dr. Vigneshwaran Muralidaran sir who gave us such an amazing research based project. We would also like to extend our gratitude to our director Shri. Manikandan sir, principal Dr. Shankar sir and chairperson Dr. Prasana Kumar sir who have always encouraged us.

MOVIE TICKET BOOKING

AIM: TO CREATE A MOVIE TICKET BOOKING SYSTEM USING MYSQL AND PYTHON.

MOTIVATION:

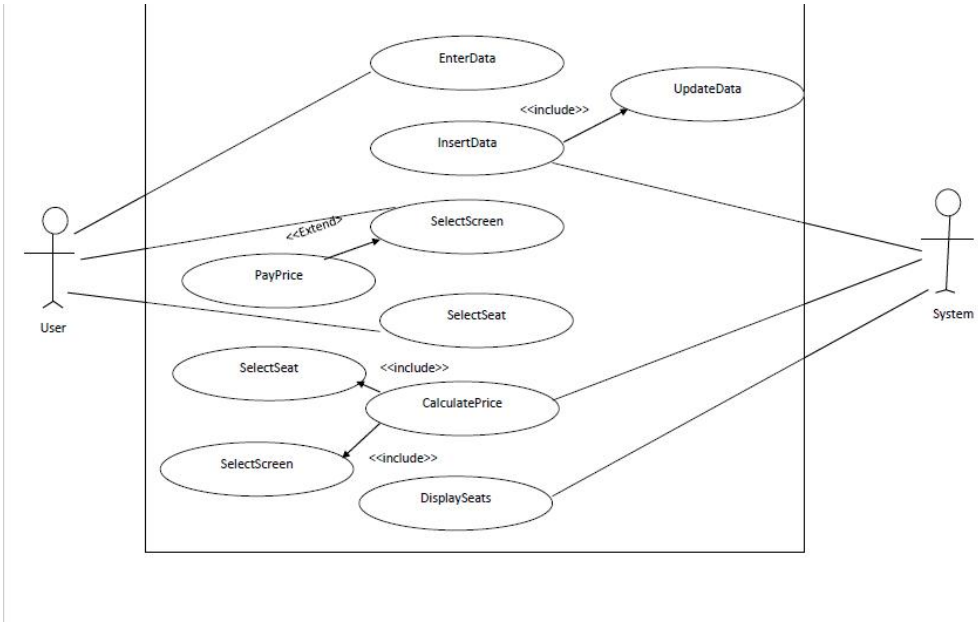
Now a days it is very common that peoples of all age groups are enjoying the cinema at the theater. It is our culture going to the theater for enjoying cinema along with the whole family. People want to spend their vacation, off-days, leisure time with their family friends and beloved persons by enjoying the cinema at the theater. No matter what ages they are.

We all are familiar in online ticket booking. We considered what is database design behind the online ticket booking system. So, we tried a basic movie ticket booking system .

Difficulties faced:

At first it was difficult to connect python with mysql as we are not familiar with that.we took help of some references like YouTube where we learnt , how to connect mysql with python.

IMPLEMENTATION:



For implementing this , there are two objects user and system,

- First user need to enter details.
- User will select the screens as there are two screens , screen1 contains less than 60 seats and screen2 consists of more than 60 seats.
- After selecting screen ,user will select seat by rows and columns that user wants.
- Then system will insert the data ,which is inserted by user after inserting the data will update into the tables .
- System will calculate price according the user selected screen and seat.
- Then finally system will display the available and vacant seats.

APPROACH

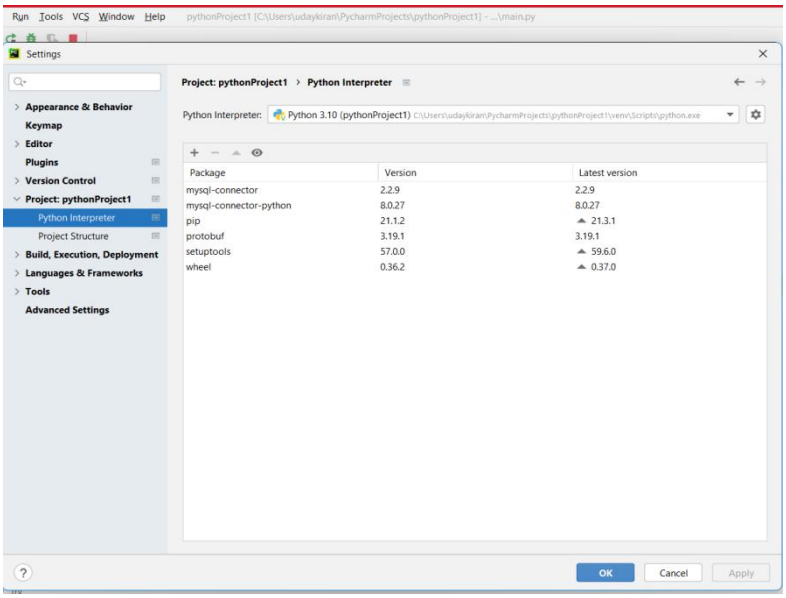
CONNECTING PYTHON AND MYSQL:

1. **Install MySQL connector module**

   Use the pip command to install MySQL connector Python.
   pip install mysql-connector-python

```
:\Windows\System32>pip install mysql-connector;
Requirement already satisfied: mysql-connector in c:\users\udaykiran\appdata\local\programs\python\python310\lib\site-packages (2.2.9)

:\Windows\System32>pip install mysql-connector-python;
Collecting mysql-connector-python
  Using cached mysql_connector_python-8.0.27-py2.py3-none-any.whl (341 kB)
Collecting protobuf>=3.0.0
  Using cached protobuf-3.19.1-py2.py3-none-any.whl (162 kB)
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.27 protobuf-3.19.1

:\Windows\System32>
```

- We used pycharm , we directly install in the mysql-connector in the project settings.
- Path:File->settings->python-interpreter and search for mysql-connector and then add the interpreter.



2. **Import MySQL connector module**

Import using a `import mysql.connector` statement so you can use this module's methods to communicate with the MySQL database.

```
import mysql.connector
from mysql.connector import Error
```

3. **Use the connect() method**

Use the `connect()` method of the MySQL Connector class with the required arguments to connect MySQL. It would return a `MySQLConnection` object if the connection established successfully.

```
connection = mysql.connector.connect(host='localhost',
                                     database='bookyourticket',
                                     user='root',
                                     password='haiiguys@1234',
                                     port='3306')
```

4. **Use the cursor() method**

Use the cursor() method of a `MySQLConnection` object to create a cursor object to perform various SQL operations.

5. **Use the execute() method**

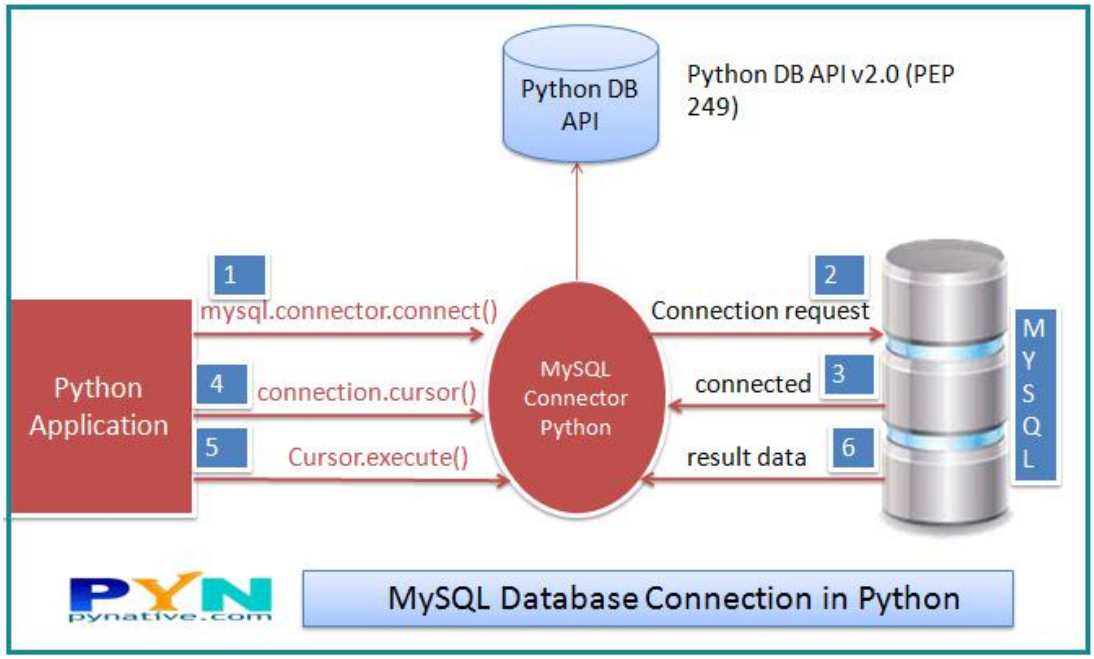
The execute() methods run the SQL query and return the result.

6. **Extract result using fetchall()**

Use `cursor.fetchall()` or `fetchone()` or `fetchmany()` to read query result.

7. **Close cursor and connection objects**

use cursor.clsoe() and connection.clsoe() method to close open connections after your work completes



TABLES IN DATABASE:

use bookyourticket; // database used.

Screen_11 table , table which stores the details of audience watching in 1st screen.

```
CREATE TABLE screen_11(
row_s11 int ,// row and the column will determine the seat number.

Col INT,
Name VARCHAR(50),
Gender VARCHAR(20),
PhoneNo VARCHAR(10),
Status VARCHAR(10),
PRIMARY KEY(row_s11,Col));
```

```
mysql> desc screen_11;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| row_s11 | int         | NO   | PRI | NULL    |       |
| Col     | int         | NO   | PRI | NULL    |       |
| Name    | varchar(50) | YES  |     | NULL    |       |
| Gender  | varchar(20) | YES  |     | NULL    |       |
| PhoneNo | varchar(10) | YES  |     | NULL    |       |
| Status  | varchar(10) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.56 sec)
```

Screen_12 table , table which stores the details of audience watching in 2nd screen.

```
CREATE TABLE screen_12(
row_s12 INT,// row and the column will determine the seat number.
Col INT,
Name VARCHAR(50),
Gender VARCHAR(20),
PhoneNo VARCHAR(10),
Status VARCHAR(10),
PRIMARY KEY(row_s12,Col));
```

```
mysql> desc screen_12;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| row_s12 | int         | NO   | PRI | NULL    |       |
| Col     | int         | NO   | PRI | NULL    |       |
| Name    | varchar(50) | YES  |     | NULL    |       |
| Gender  | varchar(20) | YES  |     | NULL    |       |
| PhoneNo | varchar(10) | YES  |     | NULL    |       |
| Status  | varchar(10) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.04 sec)
```

Statistics table will store the details of the number of tickets booked in each screen,income from each screen.

```
CREATE TABLE statistics(
Screen VARCHAR(20),
NoOfTickets INT,
Percenatge INT,//Percentage determines percent of occupancy of seats in a screen

CurrentIncome INT,//current income gives amount of income from the seats
                    booked in each screen.
TotolIncome INT);
```

```
mysql> desc statistics;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Screen        | varchar(20) | YES  |     | NULL    |       |
| NoOfTickets   | int         | YES  |     | NULL    |       |
| Percenatge    | int         | YES  |     | NULL    |       |
| CurrentIncome | int         | YES  |     | NULL    |       |
| TotolIncome   | int         | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.11 sec)
```

Python code:

we created a class bookTicket and defined class member functions.

```
1.    def insertScree_11(cls,r, c,name,gender,phoneNo,status)
2.    def insertScree_12(cls,r, c,name,gender,phoneNo,status)
3.    def updateScreen1(cls)
4.    def updateScreen2(cls)
5.    def screen1(cls,rows,cols)
6.    def screen2(cls,rows,cols)
7.    def yourChoice(cls)
8.    def screenDisp(cls)
9.   def statisticsDetail1(cls)
10.   def getUserDetail1(cls,row,col)
11.   def getUserDetail2(cls,row,col)
12.   def display(cls)
```

The above member functions are class members.

Class members:

classmethod() methods are bound to a class rather than an object. Class methods can be called by both class and object. These methods can be called with a class or with an object.

```
def display(cls)
```

When we run main.py file,it will show us this kind of interaction.

```
Hi welcome to BookYourMovie.com


**********************************************************************
Please Enter your choice:
**********************************************************************
0. Exit
1. Show the Seats.
2. Buy the ticket.
3. Statitics.
4. Show booked ticket's user Info.
Enter your choice:
```

0. exit
1. show the seats.
2. Buy the ticket.
3. Statistics.
4. Show booked ticket's user info

Display function :

```
@classmethod
  def display(cls):
    print("Please Enter your choice:")
    print("**********************************************************
**********")
    print("0. Exit")
    print("1. Show the Seats.")
    print("2. Buy the ticket.")
    print("3. Statitics.")
    print("4. Show booked ticket's user Info.")
    choice = int(input("Enter your choice:\n"))
    if choice == 0:
      exit()
    if choice == 1:
      bookTicket.yourChoice()
      bookTicket.screenDisp()

    if choice == 2:
      print("Book my ticket in:\n")
      print("1. screen1")
      print("2. Screen2")
      screenChoice = int(input("Select 1 or 2:\n"))
      if screenChoice == 1:
        r = int(input("Select the row\n"))
        range_r1 = range(1,8,1)
        if r not in range_r1:
            print("enter  between 1 and 7\n")
            r = int(input("Select the row\n"))
        c = int(input("select the col\n"))
        range_c1 = range(1,8,1)
        if c not in range_c1:
            print("enter between 1 and 7\n")
            c = int(input("select the col\n"))
        name = input("enter your name\n")
        gender = input("enter the gender\n")
        phoneNo = input("Enter the mbl no\n")
        if (len(phoneNo) != 10):
            print("phone number must be of length 10 \n")
            print("please enter the number again\n")
            phoneNo = input("Enter the mbl no\n")
        bookTicket.insertScree_11(r, c, name, gender, phoneNo, "Booked")
        bookTicket.updateScreen1()
      if screenChoice == 2:
        r = int(input("Select the row\n"))
        range_r2 = range(1, 10, 1)
```

```
        if r not in range_r2:
            print("enter  between 1 and 9\n")
            r = int(input("Select the row\n"))
        c = int(input("select the col\n"))
        range_c2 = range(1, 10, 1)
        if c not in range_c2:
            print("enter between 1 and 9\n")
            c = int(input("select the col\n"))
        name = input("enter your name\n")
        gender = input("enter the gender\n")
        phoneNo = input("Enter the mbl no\n")
        if (len(phoneNo) != 10):
            print("phone number must be of length 10 \n")
            print("please enter the number again\n")
            phoneNo = input("Enter the mbl no\n")
        bookTicket.insertScree_12(r, c, name, gender, phoneNo, "Booked")
        bookTicket.updateScreen2()
    if choice == 4:
      print("1. Show data from screen1:\n")
      print("2. Show data from screen2:\n")
      datachoice = int(input("select1 or 2:\n"))
      if datachoice == 1:
        user1Row = int(input("Enter User's Row:\n"))
        range_r1 = range(1, 8, 1)
        if user1Row not in range_r1:
            print("enter user's row between 1 and 7\n")
            user1Row = int(input("Enter User's Row:\n"))
        user1Col = int(input("Enter User's Col:\n"))
        range_c1 = range(1, 8, 1)
        if user1Col not in range_c1:
            print("enter user's column between 1 and 7\n")
            user1Col = int(input("Enter User's Col:\n"))
        bookTicket.getUserDetail1(user1Row, user1Col)
      if datachoice == 2:
        user2Row = int(input("Enter User's Row:\n"))
        range_r2 = range(1, 10, 1)
        if user2Row not in range_r2:
            print("enter user's row between 1 and 9\n")
            user2Row = int(input("Enter User's Row:\n"))
        user2Col = int(input("Enter User's Col:\n"))
        range_c2 = range(1, 10, 1)
        if user2Col not in range_c2:
            print("enter user's column between 1 and 9\n")
            user2Col = int(input("Enter User's Col:\n"))
        bookTicket.getUserDetail2(user2Row, user2Col)

    if choice == 3:
      bookTicket.statisticsDetail1()
```

1. Show the seats :

When user chooses choice 1

```
Enter your choice:
1
1.  screen1 (less than 60 seats)

2. screen2 (More than 60 seats)
```

If the option 1 is selected it shows the seating of screen1.

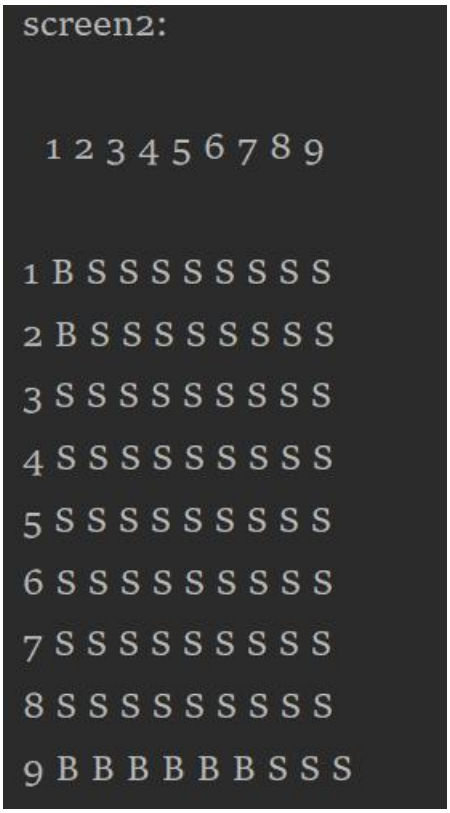
Note:reserved seats are displayed as '**B**'

Available seats are displayed as '**S**'

```
Please select the screen
1
Cinema:

screen1:

  1 2 3 4 5 6 7

1 S B S S S S S
2 B S S S S S S
3 S B S S S S S
4 B S S S B S S
5 S S S S S S S
6 S S S S S S S
7 S S S S S S S
```

In the given output 1st seat of 2nd row and 1st column is reserved(B) .

If the option 2 is selected it shows the seating of screen2.



The try block lets you test a block of code for errors.

The except block lets you handle the error.

The finally block lets you execute code, regardless of the result of the try- and except blocks.

Screen1 seating code:

```
@classmethod
 def screen1(cls,rows,cols):
     import mysql.connector
     from mysql.connector import Error

     try:
        connection = mysql.connector.connect(host='localhost',
                                             database='bookyourticket',
                                               user='root',
                                              password='haiiguys@1234',
                                               port='3306')

         sql_select_Query = "select row_s11,Col from screen_11"
         cursor = connection.cursor()
         cursor.execute(sql_select_Query)
         records = cursor.fetchall()
         print("Cinema:\n")
         print("screen1:\n")
         for m in range(0, cols):
             if m == 0:
                 print(" ", end=" ")
             else:
                 print(m, end=" ")
         for i in range(0, rows):
             print()
             for j in range(0, cols):
                 if (i == 0 and j >= 0):
                     print(" ", end=' ')
                     break
                 if i > 0 and j == 0:
                     print(i, end=' ')
                 else:
                     if (i, j) in (records):

                         print('B', end=' ')
                     else:
                         print('S', end=" ")
         print("\n*********************************************\n")
     except Error as e:
         print("Error reading data from MySQL table", e)
     finally:
         if (connection.is_connected()):
             connection.close()
             cursor.close()
```

Screen2 seating code:

```
@classmethod
  def screen2(cls,rows,cols):
      import mysql.connector
      from mysql.connector import Error

      try:
          connection = mysql.connector.connect(host='localhost',
                                               database='bookyourticket',
                                                 user='root',
                                                password='haiiguys@1234',
                                                 port='3306')

          sql_select_Query = "select row_s12,Col from screen_12"
          cursor = connection.cursor()
          cursor.execute(sql_select_Query)
          records = cursor.fetchall()
          print("Cinema:")
          print("screen2:\n")
          for m in range(0, cols):
              if m == 0:
                  print(" ", end=" ")
              else:
                  print(m, end=" ")
          for i in range(0, rows):
              print()
              for j in range(0, cols):
                  if (i == 0 and j >= 0):
                      print(" ", end=' ')
                      break
                  if i > 0 and j == 0:
                      print(i, end=' ')
                  else:
                      if (i, j) in (records):

                          print('B', end=' ')
                      else:
                          print('S', end=" ")
              print("\n****************************************\n")
      except Error as e:
          print("Error reading data from MySQL table", e)
      finally:
          if (connection.is_connected()):
              connection.close()
              cursor.close()
```

2. Buy a ticket:

If we choose option 2(buy a ticket) it will show us to select at which screen we would like to watch.

```
Enter your choice:
2
Book my ticket in:

1. screen1
2. Screen2
Select 1 or 2:
```

If we select screen1 we need to select the seat based on rows and columns, and need to enter user details (name,gender,mobile).

After booking it will display 'successfully booked'.

```
Select 1 or 2:
1
Select the row
4
select the col
4
enter your name
peter parker
enter the gender
male
Enter the mbl no
8120345671
Booked successfully
```

If we select the seat which has been already booked it will show us error.

Because row and column are combined formed a primary key.

```
Select 1 or 2:
1
Select the row
4
select the col
4
enter your name
kiran
enter the gender
male
Enter the mbl no
1234567890
Failed to insert into MySQL table 1062 (23000): Duplicate entry '4-4' for key 'screen_11.PRIMARY'
```

The above will also follows same ,if we select screen2.

The data we given will be inserted into the respective tables

Screen1 details will be inserted into screen_11 table.

Screen2 details will be inserted into screen_12 table.

Screen1 code:(inserting into the table screen_11).

```
@classmethod
  def insertScree_11(cls,r, c,name,gender,phoneNo,status):
          try:
              connection = mysql.connector.connect(host='localhost',
                                                   database='bookyourticket',
                                                         user='root',
                                                    password='haiiguys@1234',
                                                         port='3306')
              cursor = connection.cursor()
              mySql_insert_query = """INSERT INTO screen_11
(row_s11,Col,Name,Gender,PhoneNo,Status)
                                      VALUES (%s, %s, %s, %s, %s, %s)
"""

              recordTuple = (r, c, name, gender, phoneNo, status)
              cursor.execute(mySql_insert_query, recordTuple)
              connection.commit()
              print("Booked successfully ")

          except mysql.connector.Error as error:
              print("Failed to insert into MySQL table
{}".format(error))

          finally:
              if (connection.is_connected()):
                  cursor.close()
                  connection.close()
```

Screen2 code:(inserting into the table screen_12).

```
@classmethod
  def insertScree_12(cls,r, c,name,gender,phoneNo,status):
      try:
          connection = mysql.connector.connect(host='localhost',
                                               database='bookyourticket',
                                                  user='root',
                                                 password='haiiguys@1234',
                                                  port='3306')
          cursor = connection.cursor()
          mySql_insert_query = """INSERT INTO screen_12
(row_s12,Col,Name,Gender,PhoneNo,Status)
                                    VALUES (%s, %s, %s, %s, %s, %s) """

          recordTuple = (r, c,name,gender,phoneNo,status)
          cursor.execute(mySql_insert_query, recordTuple)
          connection.commit()
          print("Booked successfully ")

      except mysql.connector.Error as error:
          print("Failed to insert into MySQL table {}".format(error))

      finally:
          if (connection.is_connected()):
              cursor.close()
              connection.close()
```

- **Then system inserted the data ,which is inserted by user after inserting the data will update into the tables .**

3. Statistics

When we choose the 3rd option

It will show me the following things:

1. Number of purchased tickets.

2. Percentage of tickets booked.

3. Current income.

4. Total income.

```
Enter your choice:
3
Screen =  screen1
NoOfTickets =  7
Percentage =  14
CurrentIncome =  70
TotalIncome =  490

Screen =  screen2
NoOfTickets =  8
Percentage =  11
CurrentIncome =  80
TotalIncome =  738
```

Updating the screen1 details into statistics table:

```
@classmethod
  def updateScreen1(cls):
      import mysql.connector
      import mysql.connector
      from mysql.connector import Error

      try:
          connection = mysql.connector.connect(host='localhost',
                                               database='bookyourticket',
                                                 user='root',
                                                password='haiiguys@1234',
                                                 port='3306'
                                                 )
          cursor = connection.cursor()

          sql_update_query = """Update statistics set  NoOfTickets =
(select count(status) from screen_11),CurrentIncome=NoOfTickets *
10,Percenatge=CurrentIncome / 490 * 100 where Screen = 'Screen1'"""
          cursor.execute(sql_update_query)
          connection.commit()
      except mysql.connector.Error as error:
          print("Failed to update table record: {}".format(error))
      finally:
          if (connection.is_connected()):
              connection.close()
      bookTicket.display()
```

Updating the screen2 details into statistics table:

```
@classmethod
  def updateScreen2(cls):
      import mysql.connector
      import mysql.connector
      from mysql.connector import Error

      try:
          connection = mysql.connector.connect(host='localhost',
                                               database='bookyourticket',
                                                user='root',
                                                password='haiiguys@1234',
                                                 port='3306')
          cursor = connection.cursor()




          sql_update_query = """Update statistics set  NoOfTickets =
(select count(status) from screen_12),CurrentIncome = NoOfTickets *
10,Percenatge = CurrentIncome / 738 * 100 where Screen = 'Screen2'"""

          cursor.execute(sql_update_query)
          connection.commit()
      except mysql.connector.Error as error:
          print("Failed to update table record: {}".format(error))
      finally:
          if (connection.is_connected()):
              connection.close()
      bookTicket.display()
```

Update details:

```
 @classmethod
  def statisticsDetail1(cls):
          try:
              mySQLConnection  =
mysql.connector.connect( host='localhost',
                          database='bookyourticket',
                          user='root',
                          password='haiiguys@1234',
                          port='3306')

              cursor = mySQLConnection.cursor(buffered=True)

              sql_select_query = """select * from statistics """
              cursor.execute(sql_select_query)
              record = cursor.fetchall()

              for r in record:
                  print("Screen = ", r[0], )
                  print("NoOfTickets = ", r[1])
                  print("Percentage = ", r[2])
                  print("CurrentIncome = " ,r[3])
                  print("TotalIncome = ", r[4])
                  print()
              print("\n*****************************************\n")
          except mysql.connector.Error as error:
          # print("Failed to get record from MySQL table:
{}".format(error))
              print("Something wrong")
          finally:
              if (mySQLConnection.is_connected()):
                  cursor.close()
                  mySQLConnection.close()
          bookTicket.display()
```

4.Show booked ticket's user info

When we choose the 4th option:

It will ask the row and col number to find the user details of the booked user.

```
select1 or 2:
1
Enter User's Row:
4
Enter User's Col:
4
audience details

Name =  peter parker
Gender =  male
PhoneNo =  8120345671
Price =  $10
```

Userdetails of the screen1 for the entered row and column

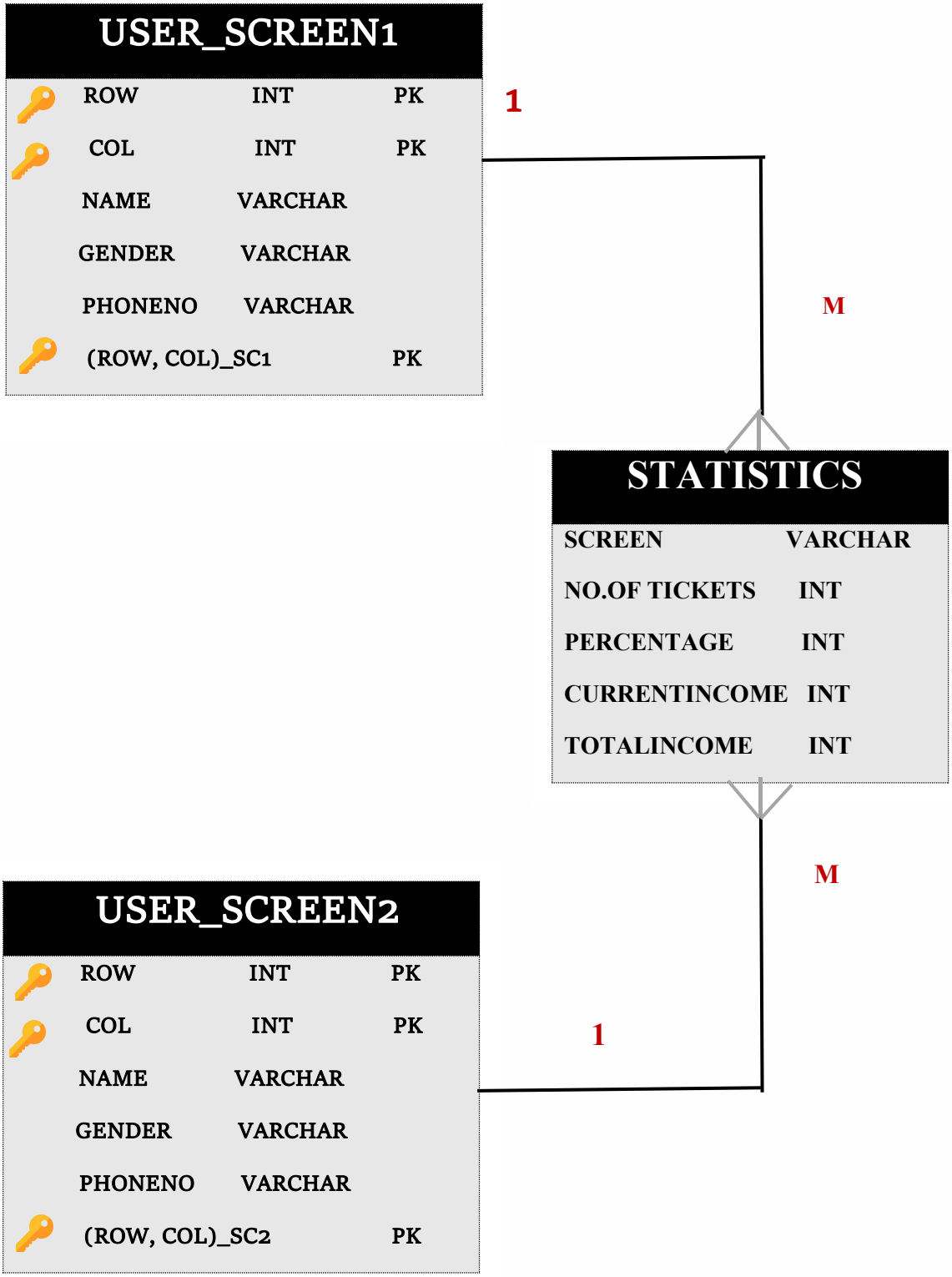
```
  @classmethod
  def getUserDetail1(cls,row,col):
      try:
          mySQLConnection = mysql.connector.connect(host='localhost',
                                                    database='bookyourticket',
                                                     user='root',
                                                     password='haiiguys@1234',
                                                      port='3306')

          cursor = mySQLConnection.cursor(buffered=True)
          sql_select_query = """select * from screen_11 where row_s11
= %s and col = %s"""
          cursor.execute(sql_select_query, (row,col))
          record = cursor.fetchall()
          print("audience details\n")
          for r in record:
              print("Name = ", r[2], )
              print("Gender = ", r[3])
              print("PhoneNo = ", r[4])
```

```
                print("Price = " ,"$10")
            print("\n*********************************************\n")
        except mysql.connector.Error as error:
            # print("Failed to get record from MySQL table:
{}".format(error))
            print("Something wrong")
        finally:
            if (mySQLConnection.is_connected()):
                cursor.close()
                mySQLConnection.close()
```

Userdetails of the screen2 for the entered row and column

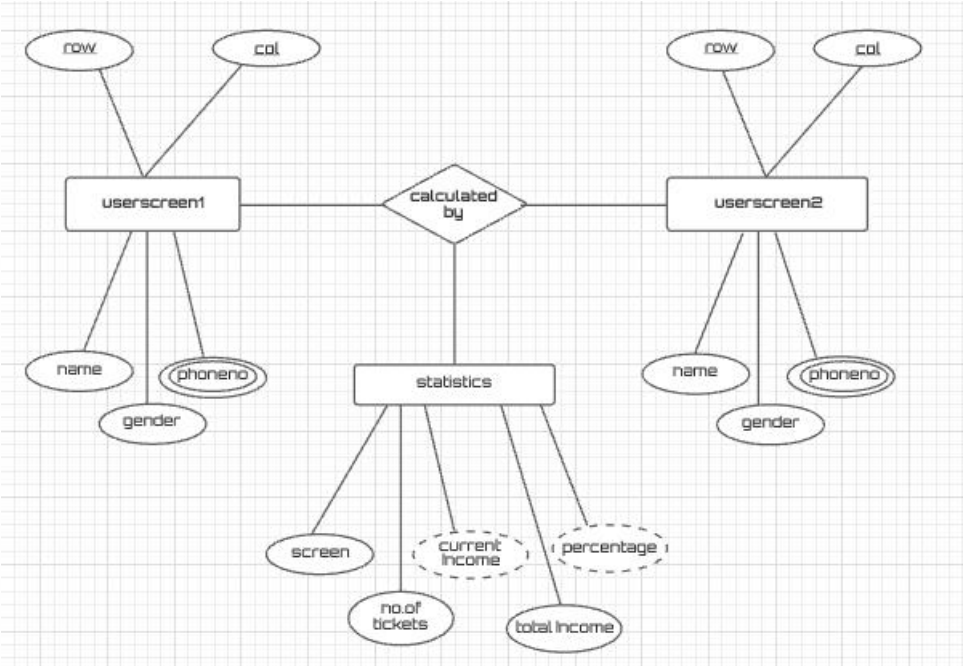
```
 @classmethod
  def getUserDetail2(cls,row,col):
        try:
            mySQLConnection = mysql.connector.connect(host='localhost',
                                                database='bookyourticket',
                                                      user='root',
                                                  password='haiiguys@1234',
                                                      port='3306')
            cursor = mySQLConnection.cursor(buffered=True)
            sql_select_query = """select * from screen_12 where row_s12
= %s and col = %s"""
            cursor.execute(sql_select_query, (row,col))
            record = cursor.fetchall()
            print("audience details:\n")
            for r in record:
                if r[0] <= 4:
                        print("Name = ", r[2])
                        print("Gender = ", r[3])
                        print("PhoneNo = ", r[4])
                        print("Price = " ,"$8")
                else:
                        print("Name = ", r[2], )
                        print("Gender = ", r[3])
                        print("PhoneNo = ", r[4])
                        print("Price = ", "$10")
            print("\n*********************************************\n")
        except mysql.connector.Error as error:
            # print("Failed to get record from MySQL table:
{}".format(error))
            print("Something wrong")

        finally:
            if (mySQLConnection.is_connected()):
                cursor.close()
                mySQLConnection.close()
```

Data base design :



IN THIS SITUATION THERE ARE TWO CASES 1. USER CAN BOOK IN ONE OR MORE SCREENS 2. USER CAN BOOK MANY SEATS IN ONE SCREEN SO, IT IS ONE TO MANY RELATIONSHIP

ER- DIAGRAM :



Er-diagram is done in lucid chart (online tool).

- Phone number is multi valued attribute (user may have more than one number).
- Current income, percentage are derived from no.of tickets so they are derived attributes.

REFERENCES:

https://pynative.com/python-mysql-database-connection/

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=video&cd=&cad=rja&uact=8&ved=2ahUKEwiR7qatl-30AhUTSmwGHUTsAX0QtwJ6BAgHEAI&url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DvR5utJvN4JY&usg=AOvVaw1SG090yDvnKHabRQoqwomn

https://beginnersbook.com/2015/04/e-r-model-in-dbms/

https://lucid.app/lucidchart/ce55d947-d3c4-479a-856d-68905df53b5c/edit?beaconFlowId=B374BE1D4995EFF0&invitationId=inv_17022832-2860-4617-8cf5-d94e625c5916&page=0_0#