

CSE2004  
Database Management Systems Project  
Resort and Hotel Management System

Sanjana D 17BCE1068  
Mukkesh G 17BCE1128  
Amrit Krishna O 17BCE1133

## **Abstract**

Our project is a Resort and Hotel Management system which helps store data, maintain records and keep essential information stored in an organised manner. As simple as this may sound, keeping track of all the data in a simple restaurant itself can by far be quite a tedious task. Extending the same to a resort and hotel makes it a much more messy task, as several other aspects such as the rooms, restaurant, facilities, reservations etc. all add on to the complexity, therefore requiring a highly efficient database management system which allows vast amounts of data to be stored, processed, manipulated and retrieved. This further enables the Resort and Hotel to maintain basic standards and offer quality service to their customers.

The front-end user will have at his service a robust user interface which would be intuitive and easy to use. It will provide a user friendly interface for data to be input and also manipulated and retrieved based on the situation and required parameters.

The site administrator will have at his service a wide set of tools which help in site development and maintenance. We will be using PyMySQL as our database management system. PyMySQL is compatible with all major web servers. It is also free and open-source thus giving it an edge in terms of security. It is also proven to be faster than most other competing database clients.

## **Module Description**

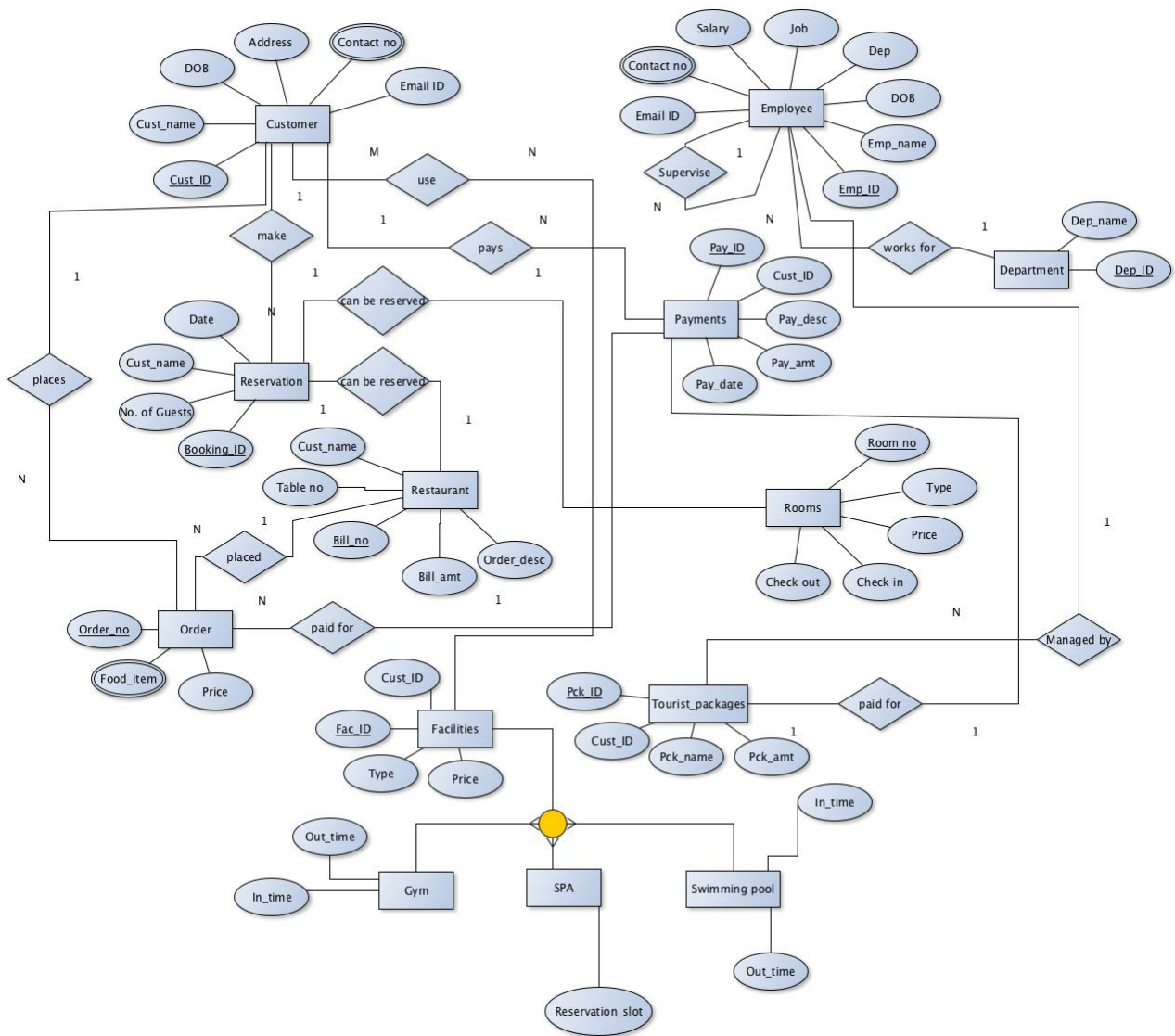
The Resort and Hotel Management system comprises of ten strong entities namely, Customer Employee, Department, Reservation, Rooms, Restaurant, Order, Payments, Facilities and Tourist Packages.

The User Registration/Login allows the user to login or register with his/her credentials. The manager can book time based slots for Facilities/Packages to be used by customers. Reservation portal allows customers to make bookings for rooms and restaurant. Restaurant table bookings can also be made by external customers and not only in-house guests at the hotel.

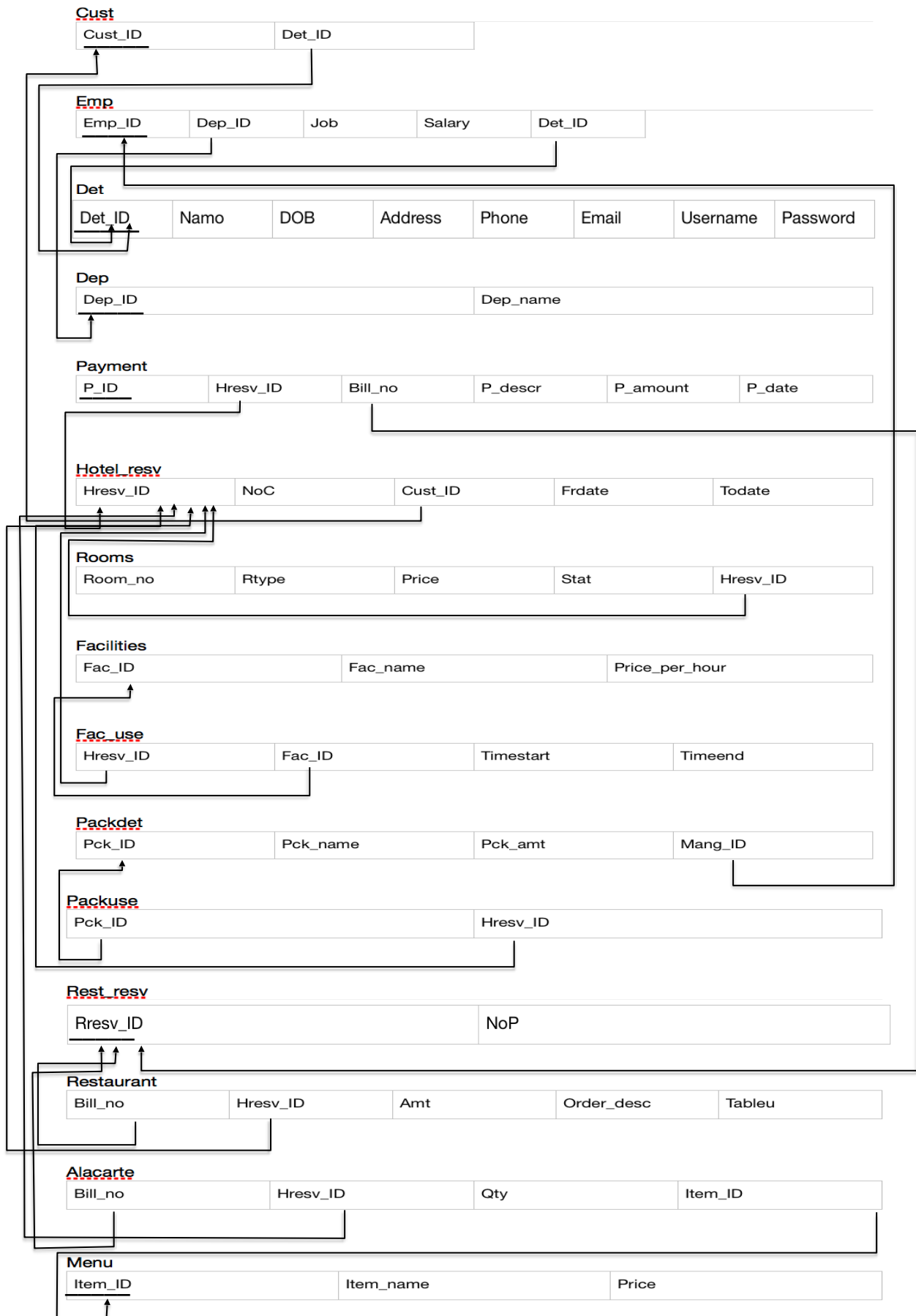
This project essentially has three user interfaces, customer, employee and admin, where each is provided with privileged access to serve their purpose and functionality. Customers once registered and logged in are allowed to make room and restaurant reservations in the hotel and also for usage of facilities and other packages offered as well. Employees can view current occupancy status of rooms and relevant details from the database and can also book facility and package slots for in-house customers. The admin is given complete database access to monitor and help maintain the system.

A conceptual model of the entire system and its components is given below.

## ER Diagram



# Relational Model Mapping



## Implementation Details

Python3 and PyMySQL were used for the complete development of this project. The database establishment and querying was done using PyMySQL while Python3.7 was used for the rest of the backend program and functional capabilities.

We also attempted to develop our project into an app with a user friendly GUI using MIT App Inventor but were not able to implement this throughout as there were too many relations and constraints to be taken care of which increased complexity and made it inconvenient.

## Hotel/Restaurant Reservation Code

```
import pymysql
db=pymysql.connect('127.0.0.1','root','Mukesh@123','resort')
cur=db.cursor()
print("\n\t\t\tReservation Portal\n1.Hotel \n2.Restaurent \nEnter: ")
choice=int(input())
while(choice==1):
    ch1=int(input("\n1.Login\n2.Register\nEnter: "))
    if(ch1==1):
        x=int(input("\n1.Customer\n2.Employee\nEnter: "))
        cur.execute('select b.log,b.pass,a.cust_id from cust a,det b where a.det_id=b.det_id')
        l=cur.fetchall()
        cur.execute('select b.log,b.pass,a.emp_id from emp a,det b where a.det_id=b.det_id')
        l1=cur.fetchall()
        u=input("\nUsername: ")
        p=input("\nPassword: ")
        while 0<1:
            flag=0
            if x==1:
                for i in l:
                    if (i[0]!='NULL' and u==i[0] and i[0]!='NULL' and p==i[1]):
                        flag=1
                        break
            elif x==2:
                for i in l1:
                    if (u!='NULL' and u==i[0] and p!='NULL' and p==i[1]):
                        flag=1
                        break
            if(flag!=1):
                print('Invalid Credentials')
                u=input("\nUsername: ")
                p=input("\nPassword: ")
            else:
                print('Successful Login')
                break

    if(x==1):
        print("\n\t\t\tHotel Reservation\n')
        cur.execute('select max(hresv_id) from hotel_resv;')
        a=int(cur.fetchall()[0][0])+1
        b=int(input('Enter the number of people: '))
        c=int(i[2])
```

```

d=input('Starting Date{YYYY-MM-DD}: ')
e=input('Ending Date{YYYY-MM-DD}: ')
cur.execute('select * from rooms where hresv_id=0;')
temp=cur.fetchall()
for i in temp:
    print(i[0],i[1],i[2])
f=int(input("\nSelect room: "))
while 0<1:
    gogo=0
    for i in temp:
        if(f==int(i[0])):
            gogo=1
            break
    if(gogo!=1):
        print("\nSelect Valid room")
    else:
        break
cur.execute("insert into hotel_resv values (%d,%d,%d,'%s','%s');"%(a,b,c,d,e))
cur.execute("update rooms set stat='occ',hresv_id=%d where room_no=%d;"%(a,f))
print("\nRoom Reserved...\nYour Reservation ID is : "+str(a))
db.commit()
elif(x==2):
    cur.execute('select * from emp,det where det.det_id=emp.det_id and emp.emp_id=%d;'%(i[2]))
    print("\nYour Details are: ")
    for i in cur.fetchall():
        print(i)

```

```

elif(ch1==2):
    cur.execute('select max(cust_id) from cust;')
    a=int(cur.fetchall()[0][0])+1
    cur.execute('select max(det_id) from det;')
    b=int(cur.fetchall()[0][0])+1
    c=input('Name: ')
    d=input('Date of birth {YYYY-MM-DD} : ')
    e=input(' Address: ')
    f=int(input('Phone: '))
    g=input('Email: ')
    h=input('Username : ')
    i=input('Password: ')
    cur.execute("insert into det values (%d,'%s','%s','%s',%d,'%s','%s','%s');"%(b,c,d,e,f,g,h,i))
    cur.execute("insert into cust values (%d,%d);"%(a,b))
    print("\nRegistration successful!")
    db.commit()
choice=int(input("\nDo you want to continue?\n1.Yes 2.No\nEnter: "))
if(choice==2):
    db.commit()
    exit()

```

```

while(choice==2):
    x=int(input("\n1.Through Hotel\n2.Direct Restaurent\nEnter: "))
    hr=0
    if(x==1):
        cur.execute('select * from rooms where hresv_id !=0 ;')
        ludo=cur.fetchall()
        for i in ludo:
            print(i)

```

```

x=int(input('\nEnter the room number: '))
hr=0
flag=1
for i in ludo:
    if(i[0]==x):
        flag=0
        hr=i[4]
        break
while(flag==1):
    print('\nInvalid Room Number')
    x=int(input('\nEnter the room number: '))
    for i in ludo:
        if(i[0]==x):
            flag=0
            hr=i[4]
            break
    cur.execute('select noc from hotel_resv where hresv_id=%d;'%(hr))
    n=int(cur.fetchall()[0][0])
    db.commit()
    print('\nTable reserved for %d people under the Hotel Reservation ID { %s }'%(n,hr))
elif(x==2):
    cur.execute('select max(rresv_id) from rest_resv;')
    od=int(cur.fetchall()[0][0])+1
    n=int(input('\nEnter the number of people: '))
    cur.execute('insert into rest_resv values (%d,%d);'%(od,n))
    print('\nReservation ID: '+str(od)+'\nNumber of people: '+str(n))
    db.commit()
choice=int(input('\nDo you want to continue?\n1.Yes\n2.No\nEnter: '))+1

```

## Packages/Facilities Usage Management Code

```

import pymysql
db=pymysql.connect('127.0.0.1','root','Mukkes@123','resort')
cur=db.cursor()
cur.execute('select b.log,b.pass,a.emp_id from emp a,det b where a.det_id=b.det_id and a.dep_id=5004')
l=cur.fetchall()
print('\n\t\t\tUsage Manager Portal\n')
u=input('\nUsername: ')
p=input('\nPassword: ')
while 0<1:
    flag=0
    for i in l:
        if (i[0]==u and p==i[1]):
            flag=1
            break
    if(flag!=1):
        print('Invalid Credentials')
        u=input('\nUsername: ')
        p=input('\nPassword: ')
    else:
        print('Successful Login')
        break
mex=1

```

```

while(mex==1):
    cur.execute('select * from rooms where hresv_id !=0 ;')
    ludo=cur.fetchall()
    for i in ludo:
        print(i)
    x=int(input('\nEnter the room number: '))
    hr=0
    flag=1
    for i in ludo:
        if(i[0]==x):
            flag=0
            hr=i[4]
            break
    while(flag==1):
        print('\nInvalid Room Number')
        x=int(input('\nEnter the room number: '))
        for i in ludo:
            if(i[0]==x):
                flag=0
                hr=i[4]
                break
    ch=int(input('\n1.Facility\n2.Package\nEnter: '))
    if(ch==1):
        a=hr
        cur.execute('select * from facilities;')
        temp=cur.fetchall()
        print(temp,sep='\n')
        b=int(input('\nSelect the facility ID: '))
        pogo=1
        while (pogo==1):
            for i in temp:
                if b==i[0]:
                    pogo=0
                    break
            if(pogo!=0):
                b=int(input("\nSelect the facility ID: "))
        c=input("\nEnter start time {HH:MM:SS} : ")
        d=input("\nEnter end time {HH:MM:SS} : ")
        cur.execute("insert into fac_use values (%d,%d,'%s','%s');"%(a,b,c,d))
        db.commit()
        print("\nSuccessfully Logged. Authorised to enter.")
    elif(ch==2):
        b=hr
        cur.execute('select * from packdet;')
        temp=cur.fetchall()
        for i in temp:
            print(i)
        a=int(input("\nSelect the Packet ID: "))
        while 0<1:
            chomp=0
            for i in temp:
                if(i[0]==a):
                    chomp=1
                    break
            if(chomp!=1):
                a=int(input("\nSelect the Packet ID: "))

```



```

else:
    break
cur.execute('insert into packuse values(%d,%d);'%(a,b))
db.commit()
print('\nSuccessfully booked. Package Included.')
mex=int(input('\nDo you want to continue?\n1.Yes\n2.No\nEnter: '))

```

## Results

### Reservation Module screenshots

```

*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\MukKesh McKenzie\Desktop\DBMS workkk\work.py =====

                        Reservation Portal

1.Hotel
2.Restaurent
Enter:
2

1.Through Hotel
2.Direct Restaurent
Enter: 2

Enter the number of people: 5

Reservation ID: 212
Number of people: 5

Do you want to continue?
1.Yes
2.No
Enter: 1

1.Through Hotel
2.Direct Restaurent
Enter: 1
(402, 'double', 1200.0, 'occ', 100)
(406, 'dub-deluxe', 1500.0, 'occ', 101)
(403, 'triple', 1600.0, 'occ', 102)
(405, 'tri-deluxe', 1800.0, 'occ', 103)
(404, 'single', 800.0, 'occ', 104)
(401, 'single', 800.0, 'occ', 105)
(412, 'tri-deluxe', 1800.0, 'occ', 106)

Enter the room number: 401

Table reserved for 1 people under the Hotel Reservation ID { 105 }

Do you want to continue?
1.Yes
2.No
Enter: |
Ln: 44 Col: 7

```

Reservation Portal

1.Hotel  
2.Restaurent  
Enter:  
1

1.Login  
2.Register  
Enter: 1

1.Customer  
2.Employee  
Enter: 1

Username: brohit  
  
Password: fornite  
Successful Login

Hotel Reservation

Enter the number of people: 4  
Starting Date(YYYY-MM-DD): 2018-12-12  
Ending Date(YYYY-MM-DD): 2018-12-13  
407 dub-deluxe 1500.0  
408 tri-deluxe 1800.0  
409 sin-deluxe 1000.0  
410 dub-deluxe 1500.0  
411 dub-deluxe 1500.0  
413 single 800.0  
414 double 1200.0

Select room: 408

Room Reserved...  
Your Reservation ID is : 107

Do you want to continue?  
1.Yes 2.No  
Enter: |

Ln: 42 Col: 0

Reservation Portal

1.Hotel  
2.Restaurent  
Enter:  
1

1.Login  
2.Register  
Enter: 1

1.Customer  
2.Employee  
Enter: 2

Username: sanjana

Password: anole  
Successful Login

Your Details are:  
(8002, 5003, 'Package Manager', 25000.0, 9003, 9003, 'sanjana', datetime.date(1999, 8, 17), 'Panache, OMR', 24727985, 'sanjago@gmail.com', 'sanjana', 'anole')

Do you want to continue?  
1.Yes 2.No  
Enter: 2

```

                                Reservation Portal

1.Hotel
2.Restaurent
Enter:
1

1.Login
2.Register
Enter: 2
Name: navya
Date of birth {YYYY-MM-DD} : 1999-06-28
Address: The Divine,Nungambakkam
Phone: 28171900
Email: navzdacoollest@gmail.com
Username : navz
Password: kewlz

Registration successful!

Do you want to continue?
1.Yes 2.No
Enter: 1

1.Login
2.Register
Enter: 1

1.Customer
2.Employee
Enter: 1

Username: navz

Password: kewlz
Successful Login

```

## Package/Facility Usage Module Screenshots

```

                                Usage Manager Portal

Username: mahesh

Password: fluffy
Successful Login
(402, 'double', 1200.0, 'occ', 100)
(406, 'dub-deluxe', 1500.0, 'occ', 101)
(403, 'triple', 1600.0, 'occ', 102)
(405, 'tri-deluxe', 1800.0, 'occ', 103)
(404, 'single', 800.0, 'occ', 104)
(401, 'single', 800.0, 'occ', 105)
(412, 'tri-deluxe', 1800.0, 'occ', 106)
(408, 'tri-deluxe', 1800.0, 'occ', 107)

Enter the room number: 404

1.Facility
2.Package
Enter: 1
((6001, 'Gym', 150.0), (6002, 'Spa', 250.0))

Select the facility ID: 6001

Enter start time {HH:MM:SS} : 12:00:00

Enter end time {HH:MM:SS} : 13:00:00

Successfully Logged. Authorised to enter.

Do you want to continue?
1.Yes
2.No
Enter: |

```

## Usage Manager Portal

Username: mahesh

Password: fluffy

Successful Login

```
(402, 'double', 1200.0, 'occ', 100)
(406, 'dub-deluxe', 1500.0, 'occ', 101)
(403, 'triple', 1600.0, 'occ', 102)
(405, 'tri-deluxe', 1800.0, 'occ', 103)
(404, 'single', 800.0, 'occ', 104)
(401, 'single', 800.0, 'occ', 105)
(412, 'tri-deluxe', 1800.0, 'occ', 106)
(408, 'tri-deluxe', 1800.0, 'occ', 107)
```

Enter the room number: 408

1.Facility

2.Package

Enter: 2

```
(2001, 'Kodaikanal', 8000.0, 8002)
```

```
(2002, 'Ooty', 10000.0, 8002)
```

Select the Packet ID: 2001

Successfully booked. Package Included.

Do you want to continue?

1.Yes

2.No

Enter: 2

## Instance Screenshots

	item_id	item_name	price
1	7001	Soup	60
2	7002	Garlic Bread	75
3	7003	Ice Cream	80

	det_id	namo	DOB	address	phone	email	log	pass
1	0	NULL	1200-12-12	NULL	0	NULL	NULL	NULL
2	9001	mukresh	1999-08-12	32,Lakshmi nagar, chromepet	22414235	mukmckenzie@gmail.com	mex	123
3	9002	amrit	1999-09-30	21G, Rainbow colony, OMR	22652425	amrit.om@outlook.com	amrit	doggo
4	9003	sanjana	1999-08-17	Panache, OMR	24727985	sanjago@gmail.com	sanjana	anole
5	9004	rohit	1999-07-30	21G, Rainbow colony, Oragadam	22652355	rohit@outlook.com	brohit	fornite
6	9005	ganesh	1972-04-19	45,mux nagar, dmux	27653948	gaanesh@gmail.com	gani	kavi
7	9006	kavitha	1980-06-26	32,luxmi nagar, chromepet	22652425	kavithathehr@yahoo.co.in	kavi	muki
8	9007	mahesh	1999-09-20	3B,Jains inseli,vadapalani	22412355	mahesh@yahoo.com	mahesh	fluffy
9	9008	erik	1999-09-09	opaline,lobby,omr	98658765	erik@gmail.com	erik	amrit
10	9009	naveen	1999-08-09	32,justin park	67453452	naveen@gmail.com	naveen	winner
11	9010	navya	1999-06-28	The Divine,Nungambakkam	28171900	navzdacoollest@gmail.com	navz	kewlz

	dep_id	dep_name
1	5001	cleaning
2	5002	lobby
3	5003	Package
4	5004	facuse

←	cust	↺	⬇
---	------	---	---

	cust_id	det_id
1	0	0
2	1001	9001
3	1002	9004
4	1003	9005
5	1004	9006
6	1005	9008
7	1006	9009
8	1007	9010

←	hotel_resv
---	------------

	hresv_id	noc	cust_id	frdate	todate
1	0	0	0	1200-12-12	1200-12-12
2	100	3	1001	2018-08-25	2018-08-27
3	101	2	1002	2018-12-12	2018-12-13
4	102	4	1003	2018-12-03	2018-12-04
5	103	3	1004	2019-01-01	2019-01-05
6	104	1	1002	2018-11-12	2018-11-13
7	105	1	1005	2018-11-12	2018-11-14
8	106	3	1006	2018-11-12	2018-11-13
9	107	4	1002	2018-12-12	2018-12-13

←	rooms	↺	⬇
---	-------	---	---

	room_no	rtype	price	stat	hresv_id
1	401	single	800	occ	105
2	402	double	1200	occ	100
3	403	triple	1600	occ	102
4	404	single	800	occ	104
5	405	tri-deluxe	1800	occ	103
6	406	dub-deluxe	1500	occ	101
7	407	dub-deluxe	1500	emp	0
8	408	tri-deluxe	1800	occ	107
9	409	sin-deluxe	1000	emp	0
10	410	dub-deluxe	1500	emp	0
11	411	dub-deluxe	1500	emp	0
12	412	tri-deluxe	1800	occ	106
13	413	single	800	emp	0
14	414	double	1200	emp	0

←	facilities	↺	⬇
---	------------	---	---

	fac_id	fac_name	price_per_hour
1	6001	Gym	150
2	6002	Spa	250

←	fac_use	↺	⬇
---	---------	---	---

	hresv_id	fac_id	timestart	timeend
1	102	6002	08:30:00	09:30:00
2	101	6002	09:30:00	10:30:00
3	100	6001	05:30:00	07:00:00
4	102	6002	08:30:00	10:00:00
5	103	6001	08:30:00	09:30:00
6	104	6001	12:00:00	13:00:00

## ← packdet



	pck_id	pck_name	pck_amt	mang_id
1	2001	Kodaikanal	8000	8002
2	2002	Ooty	10000	8002

## ← packuse

	pck_id	hresv_id
1	2002	103
2	2002	103
3	2001	101
4	2001	105
5	2002	105
6	2001	107

## ← rest\_resv



	rresv_id	nop
1	0	0
2	207	5
3	208	5
4	209	4
5	210	6
6	211	5
7	212	5

## ← payment

	p_id	hresv_id	bill_no	p_descr	p_amount	p_date
1	701	0	207	Garlic+Soup	235	2018-11-12

## ← emp

	emp_id	dep_id	job	salary	det_id
1	8001	5002	Lobby manager	25000	9002
2	8002	5003	Package Manager	25000	9003
3	8003	5004	Usage manager	20000	9007

## ← alacarte

	bill_no	hresv_id	item_id	qty
1	207	0	7001	2
2	207	0	7002	1

## ← restaurent

	bill_no	hresv_id	amt	order_desc	tableu
1	207	0	235	Garlic+Soup	7

## Conclusion

After multiple iterations and constant refining we developed our software product, a Resort and Hotel Management System. Several hurdles were faced during the entire development process but alternative solutions were found and implemented in order to get past them and develop this product.

We encountered problems right from the primary stage of inputting data values into our database as small errors such as 'Parent Key Not Found' and 'Database Locked' held us back for quite a while until we got the hang of it. Due to a high number of relations in our database and the numerous constraints present we found it difficult to develop the front end GUI with the appropriate connections and functionalities using a python module TKinter. The number of parameters which had to be taken into account greatly increased the complexity forcing us to change our development platform and modules. We also attempted to use another development tool called MIT App Inventor to migrate our program to an android application however again complexity issues came into place.

Thus, we resorted to using the standard python shell to develop our program along with PyMySQL for the entire database management. The program can now be used to conveniently store, process, manipulate and retrieve data in the appropriate environment.

## References

1. <https://www.tutorialspoint.com/python3/index.htm>
2. <https://pymysql.readthedocs.io/en/latest/>
3. <https://dev.mysql.com/doc/refman/8.0/en/>