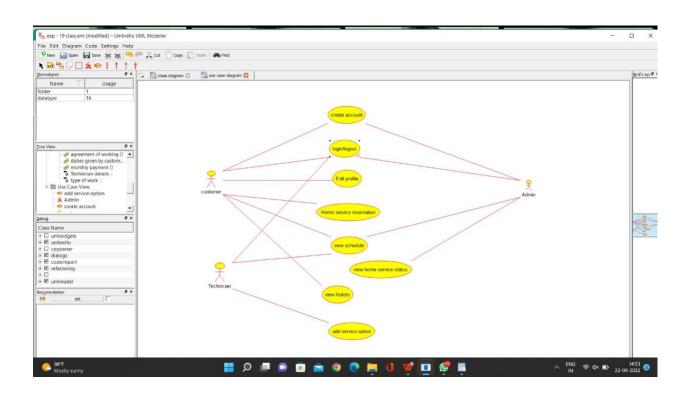
Name: B. Aravinda Valli

Reg No:192011442

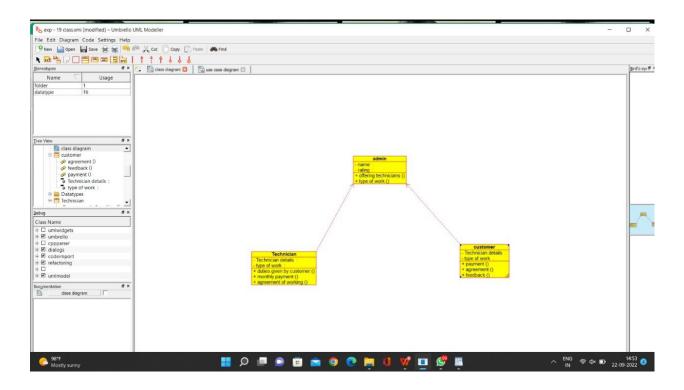
Experiment-19

Home service system

USE CASE DIAGRAM:



CLASS DIAGRAM:



#

coding=Sy

stem

class

admin(ob

ject):

:version:		
:author:		
шш		
ппп		
ATTRIBUT		
ES		
name_		
(private)		
rating_		
(private)		
нин		

```
def
offering_t
echnician
s_(self):
  111111
@return
@author
  111111
  pass
def
type_of_
work_(sel
f):
  111111
```

@return :
@author
11111
pass
#
coding=Sy
stem
from
PAYMENT
import *
class
BANK(obj
ect):

:version:
:author:
ATTRIBUT
ES
account_
no_
(private)
account_t
ype_
(private)
#
coding=Sy
stem

from
PAYMENT
import *
class
CASH_ON
_DELIVER
Y(object):
ппп
:version:
:author:
.autilor.
ппп
нин
ATTRIBUT
ES

```
customer
_name_
(private)
customer
_address
(private)
\mathsf{amount}\_
(private)
 .....
def
updates_t
he_mode
_of_paym
```

ent_(self)
:
ппп
@return
:
@author

pass
#
coding=Sy
coding=Sy
coding=Sy stem
coding=Sy stem from
coding=Sy stem from SYSTEM
coding=Sy stem from SYSTEM import *
coding=Sy stem from SYSTEM import *
coding=Sy stem from SYSTEM import * from RESTAUR

нин		
:version:		
:author:		
ппп		
11111		
ATTRIBUT		
ES		
ID_		
(private)		
ппп		

class

CUSTOME

R(object):

```
def
login_(sel
f):
  111111
@return
@author
  111111
  pass
def
select_res
taurant_(
self):
  111111
```

@return

•

```
@author
  .....
  pass
def
place_ord
er_(self):
  111111
@return
@author
  .....
  pass
def
payment_
(self):
  111111
```

@author pass # coding=Sy stem from PERMISSI ON_CLAS S import class CUSTOME R_CLASS (PERMISSI ON_CLAS S):

@return

ппп
:version:
:author:
ATTRIBUT
ES
ID
(private)

name

(private)

mobile_a nd_addre SS (private) def add_cust omer_(sel f): 111111 @return @author 111111

pass

```
def
edit_cust
omer_(sel
f):
  .....
@return
@author
  111111
  pass
def
delete_cu
stomer_(s
elf):
  111111
```

@return	
:	
@author	
ппп	
pass	
def	
search_cu	
stomer_(s	
elf):	
ппп	
@return	
:	
@author	
ппп	
pass	

#	
coding=Sy	
stem	
class	
DRIVING_	
LICENSE_	
CLASS(obj	
ect):	
11111	
:version:	
:author:	
idatilori	
11111	
ATTRIBUT	
ES	

type_
(private)
name_
(private)
. ,
ID
(private)
def
add_drivi
ng_licens
e_(self):

@return	
:	
@author	
нин	
pass	
def	
edit_drivi	
ng_licens	
e_(self):	
@return	
:	
@author	
ппп	
pass	

```
def
\mathsf{delete\_dr}
iving_lice
nse_(self)
  .....
@return
@author
  111111
  pass
def
search\_dr
iving_lice
nse_(self)
  111111
```

@author			
pass			
#			
coding=Sy			
stem			
class			
DRIVING_			
LICENSE_			
CLASS(obj			
ect):			
ппп			
:version:			
:author:			

@return

ID (private)

.....

```
add_drivi
ng_licens
e_(self):
 .....
@return
@author
 111111
 pass
def
edit_drivi
ng_licens
e_(self):
```

def

```
@return
@author
 .....
 pass
def
delete_dr
iving_lice
nse_(self)
 111111
@return
@author
 111111
 pass
```

```
def
search\_dr
iving_lice
nse_(self)
 .....
@return
@author
 111111
 pass
coding=Sy
stem
class
PERMISSI
ON_CLAS
S(object):
```

:version:	
:author:	
11111	
ATTRIBUT	
FC	
ES	
ID_	
(private)	
role_	
(private)	
,	

```
\mathsf{title}\_
(private)
 .....
def
add_per
mission_(
self):
  .....
@return
@author
  111111
  pass
 def
edit_per
```

```
self):
  .....
@return
@author
  .....
  pass
def
delete_pe
rmission_
(self):
  .....
@return
```

mission_(

```
@author
  .....
  pass
def
search_p
ermission
_(self):
  .....
@return
@author
  .....
  pass
coding=Sy
stem
from
```

PERMISSI

ON_CLAS
S import
*
class
REGISTRA
TION_CLA
SS
(PERMISSI
ON_CLAS
S):
:version:
:author:
.uutiloi.
ATTRIBUT
ES

ID_
(private)
customer
_
(private)
number_
(private)
. ,
name
(private)

```
type_
(private)
 111111
def
add_regis
tration_(s
elf):
  .....
@return
@author
  .....
  pass
def
edit_regis
tration_(s
elf):
```

@return

:

@author

111111

pass

def

delete_re

gistration

_(self):

.....

@return

•

@author

```
pass
```

def
search_re
gistration
_(self):

.....

@return

@author

"""

pass

coding=Sy stem

ROLE_CL
ASS(objec
t):
"""
:version:
:author:
ппп
нин
ATTRIBUT
ES
ID
(private)

class

title_
(private)
descriptio
n_ (arisata)
(private)
def
add_role_
(self):
@return

@author

pass def edit_role _(self): @return @author 111111 pass def delete_ro

le_(self):

111111

@return
:
@author
ппп
pass
def
search_ro
le_(self):
@return
:
@author
11111
pass

```
assign_rol
e_(self):
  111111
@return
@author
  111111
  pass
coding=Sy
stem
class
USER(obj
ect):
 111111
```

def

:version:
:author:
нин
ATTRIBUT
ES
ID_
(private)
шш
def
name_(se
If):

@return	
:	
@author	
"""	
pass	
def	
email_(se	
If):	
пип	
@return	
:	
@author	
11111	
pass	

password
_(self):
нин
@return
:
@author
нин
pass
def
Cv_uploa
d_(self):
"""

@return

def

```
@author
  .....
  pass
def
mobile_n
umber_(s
elf):
  .....
@return
@author
  .....
  pass
coding=Sy
stem
from
```

PERMISSI

ON_CLAS
S import
*
class
VEHICLE_
CLASS(obj
ect):
"""
:version:
:author:
шш
ATTRIBUT
ES

Id
(private)
type_
(private)
number_
(private)
,
customer
id
(private)
(private)

```
def
add_vehi
cle_(self):
  111111
@return
@author
  111111
  pass
def
edit_vehi
cle_(self):
  .....
```

@return

```
@author
 .....
 pass
def
delete_ve
hicle_(self
):
 .....
@return
@author
 .....
 pass
def
search_ve
hicle_(self
):
```

@return

:

@author

.....

pass

#

coding=Sy

stem

from

PERMISSI

ON_CLAS

S import

*

class

VEHICLE_

TYPE_CLA

SS

(PERMISSI

ON_CLAS		
S):		
пип		
#		
coding=Sy		
stem		
Class		
PAYMENT		
(object):		
нин		
:version:		
:author:		
шш		
шш		
ATTRIBUT		

ES

```
\mathsf{amount}\_
(private)
payment_
option_
(private)
 .....
def
accept_th
e_payme
nt_(self):
```

```
@return
@author
 .....
 pass
def
process_t
he_paym
ent_(self)
 111111
@return
@author
 111111
 pass
```

```
def
delete\_th
e_payme
nt(self):
 .....
@return
@author
 111111
 pass
coding=Sy
stem
class
RESTAUR
ANT(obje
ct):
```

:version:	
:author:	
ATTRIBUT	
ES	
name_	
(private)	
11111	
def	
check_th	
e_order_(
self):	
111111	

@return
:
@author
11111
pass
def
delivered
_the_ord
er_(self):
@return
:
@author
pass

coding=Sy			
stem			
from			
PAYMENT			
import *			
class			
SODEXO_			
SYSTEM(o			
bject):			
11111			
:version:			
:author:			
.autiloi.			
ATTRIBUT			
ES			

#

CouponN 0 (private) def accept_co upon_(sel f): @return @author

pass

```
reject_co
upon_(sel
f):
  111111
@return
@author
  111111
  pass
#
coding=Sy
stem
from
PAYMENT
import *
```

def

:version:			
:author:			
шш			
ATTRIBUT			
ES			
order_			
(private)			

class

SYSTEM(o

bject):

menu_car ds_ (private) restauran t_list_ (private) customer $_id_pass$ word_ (private)

payment_

```
option_
(private)
111111
def
updates_
order\_sta
tus_(self):
 .....
@return
@author
 .....
 pass
def
updates_
payment_
```

```
status_(s
elf):
  .....
@return
@author
  111111
  pass
def
updates_
delivery_s
tatus_(sel
f):
  111111
```

@return

:

```
@author
```

pass

```
online_pa
yment_Id
_passwor
d_
(private)
```

111111

```
def
accept_th
e_payme
nt_(self):
"""
```

@return @author 111111 pass def reject_th e_payme nt_(self): @return @author

pass

#			
coding=Sy			
stem			
from			
admin			
import *			
·			
class			
customer			
(object):			
:version:			
:author:			
11111			
ппп			
ATTRIBUT			

ES

n_details (private) type_of_ work_ (private) 111111 def payment_ (self):

111111

Technicia

@return
:
@author
11111
pass
def
agreemen
t_(self):
@return
:
@author
ппп
pass

def
feedback
_(self):
111111
@return
:
@author
11111

pass