



Automated Toll Collection System

Advance Database Management System

Section: C

GUIDED BY

Rezwan Ahmed

Department of Computer Science

American International University-Bangladesh

PREPARED BY

| NO | NAME | ID |
|----|------------------|------------|
| 1. | FARUK, AL MOMIN | 16-32369-2 |
| 2. | AHMED, RAFI | 16-32677-3 |
| 3. | RAHMAN, SHOJIBUR | 18-36205-1 |
| 4. | NOOR, ZUHAN | 18-36515-1 |

CONTENTS

| | |
|----------------------------|----|
| 1. Introduction: | 2 |
| 2. User Category: | 2 |
| 3. Feature List: | 2 |
| 4. ER Diagram: | 3 |
| 5. Class Diagram: | 4 |
| 6. Use Case Diagram: | 5 |
| 7. Activity Diagram: | 6 |
| 8. Table Creation: | 7 |
| 8.1. UserS: | 7 |
| 8.2. Toll: | 7 |
| 8.3. Vehicle: | 8 |
| 8.4. Transaction: | 8 |
| 9. Table Data: | 9 |
| 9.1. UserS: | 9 |
| 9.2. Vehicle: | 9 |
| 9.3. Toll: | 10 |
| 9.4. Transaction | 10 |
| 10. View | 11 |
| 11. Procedure | 12 |
| 12. Function | 13 |
| 13. Trigger..... | 16 |

1. INTRODUCTION:

The automated toll collection system will work as a prepaid toll payment system which will save a lot of time since vehicles passing through the toll plaza won't have to stop to pay the toll anymore. Our country's toll collection system, is one of the sole reasons for the traffic jam on highways and we hope to offer a solution for this problem via this project.

2. USER CATEGORY:

There are two types of Users here. They are:

- Admin
- User

3. FEATURE LIST:

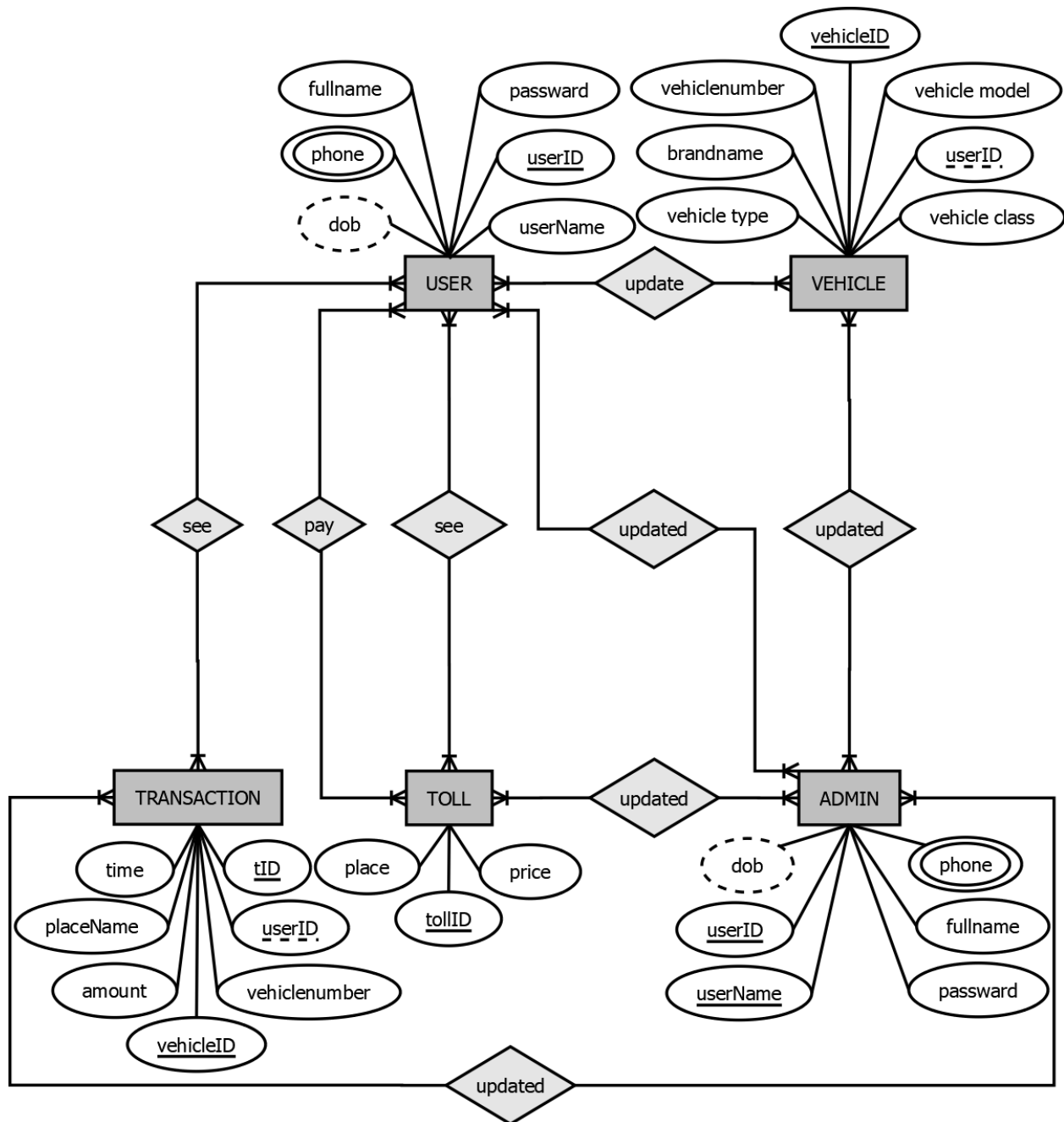
In this project the “Admin” has the following features:

- Introducing different toll packages
- Managing those toll packages
- Checking reports of all toll transactions
- Checking existing user lists

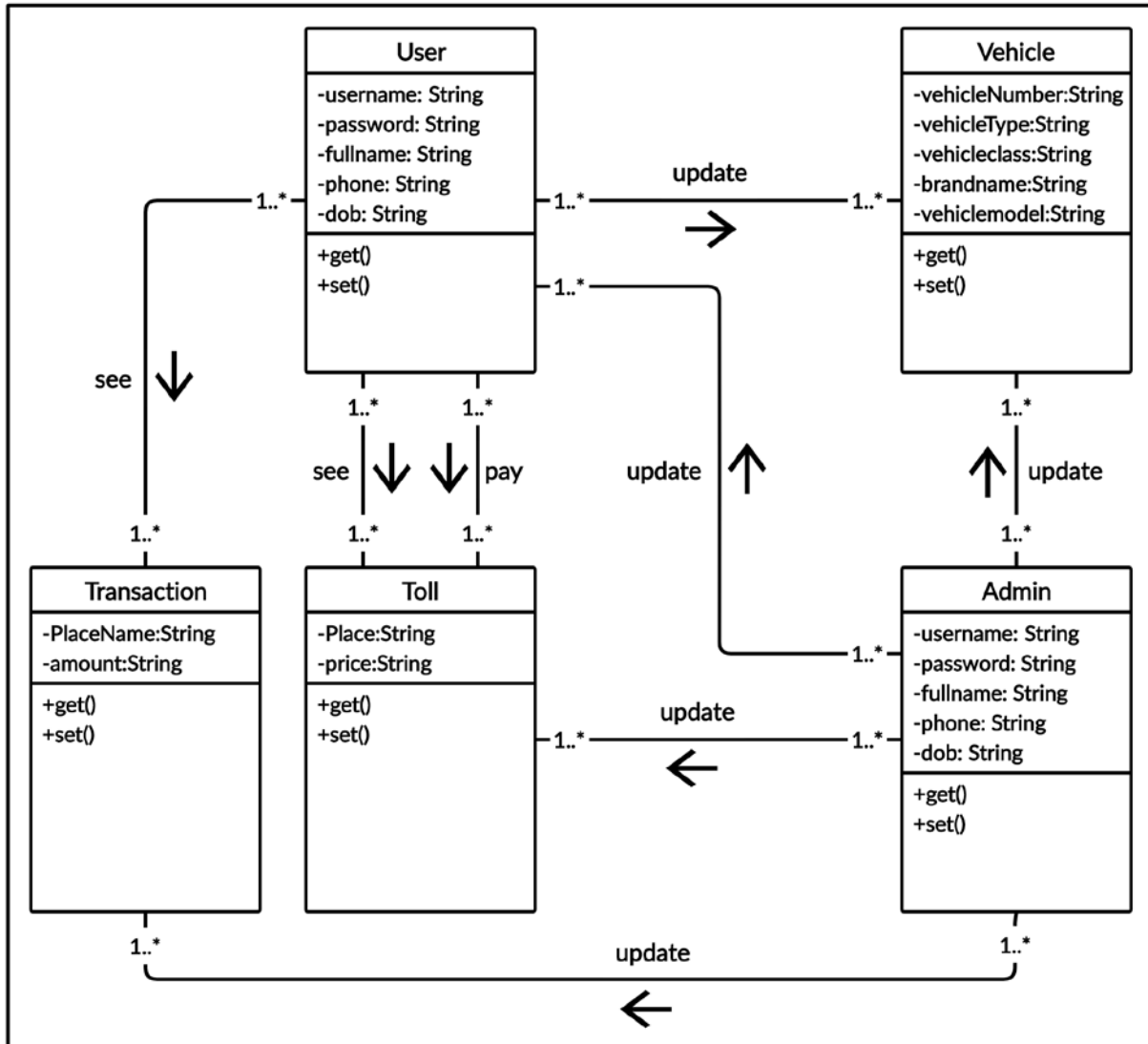
In this project the “User” has the following features:

- Profile Creation
- Adding unlimited vehicles
- Managing existing vehicles
- Checking self-transaction history
- Checking all available toll places and their fares
- Paying the toll before crossing the gate

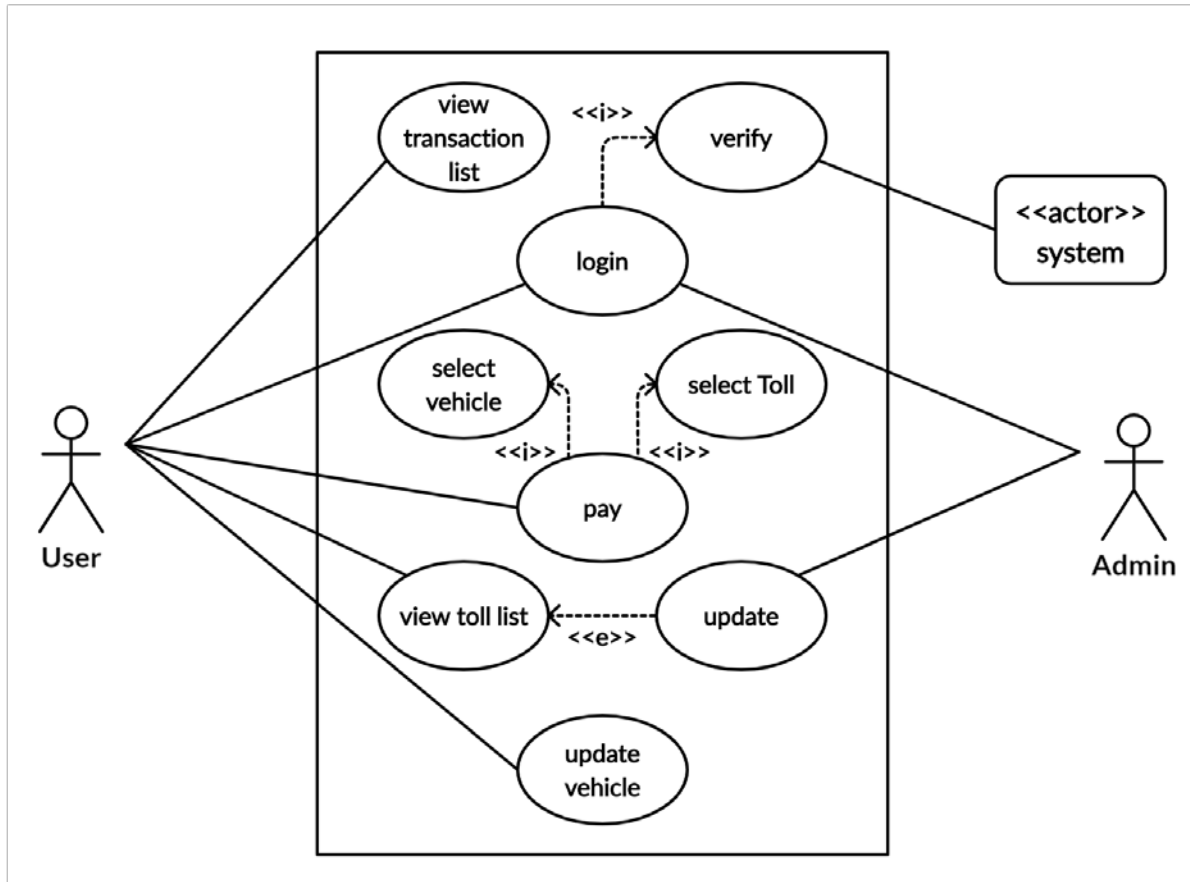
4. ER DIAGRAM:



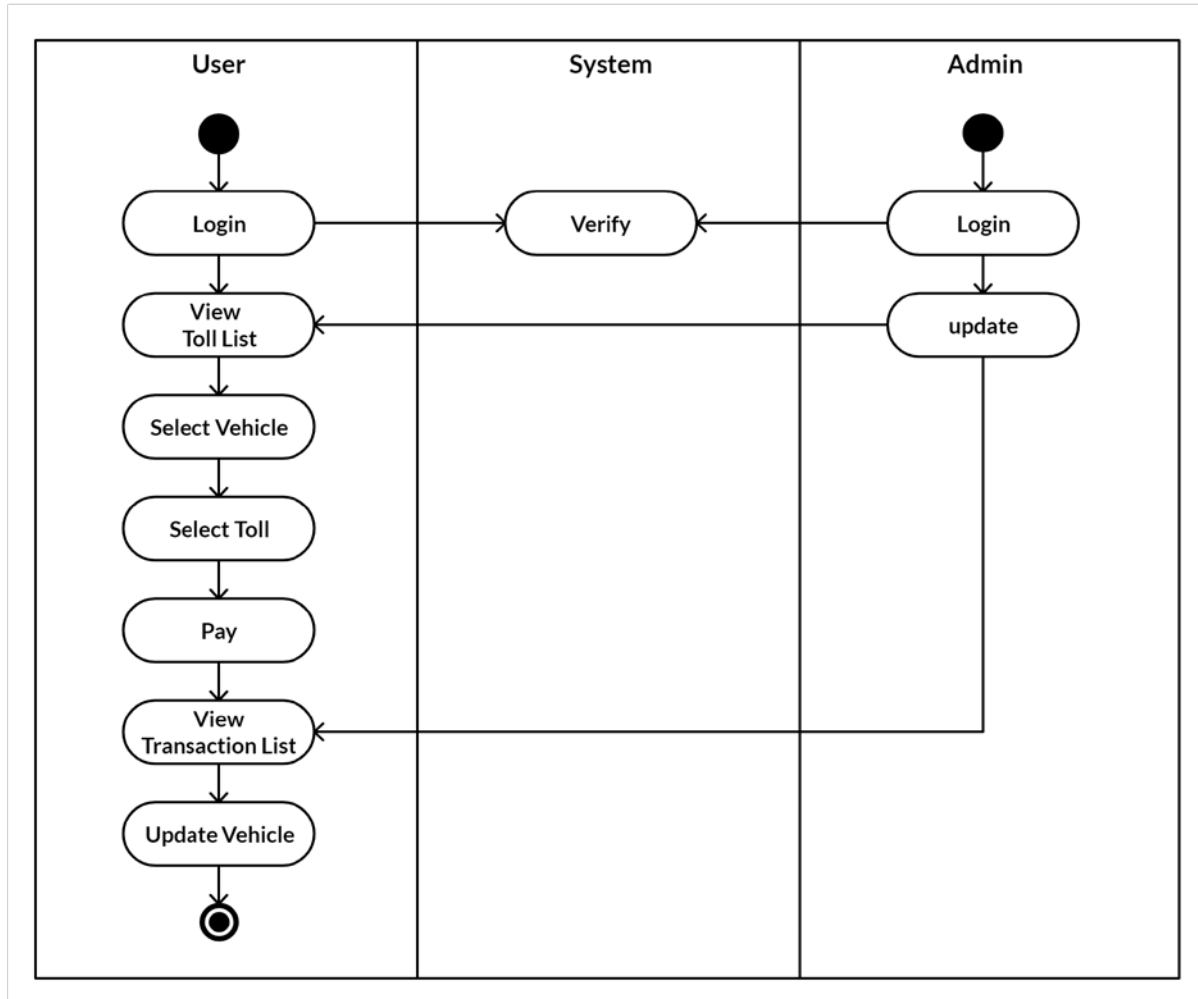
5. CLASS DIAGRAM:



6. USE CASE DIAGRAM:



7. ACTIVITY DIAGRAM:



8. TABLE CREATION:

8.1. USERS:

```
CREATE TABLE users (  
    userid number(11) NOT NULL PRIMARY KEY,  
    username varchar2(50) NOT NULL ,  
    password varchar2(50) NOT NULL,  
    fullname varchar2(50) NOT NULL,  
    phone varchar2(50) NOT NULL,  
    dob varchar2(50),  
    admin number(11),  
    CONSTRAINT username_unique UNIQUE (username)  
);  
  
CREATE SEQUENCE seq_users MINVALUE 1 START WITH 1 INCREMENT BY 1  
CACHE 10;
```

8.2. TOLL:

```
CREATE TABLE toll (  
    tollID number(11) NOT NULL PRIMARY KEY,  
    place varchar2(50) NOT NULL,  
    price number(11) NOT NULL,  
    CONSTRAINT place_unique UNIQUE (place)  
);  
  
CREATE SEQUENCE seq_toll MINVALUE 1 START WITH 1 INCREMENT BY 1  
CACHE 10;
```


8.3. VEHICLE:

```
CREATE TABLE vehicle(  
    vehicleid number(11) NOT NULL PRIMARY KEY,  
    brandname varchar(30) NOT NULL,  
    vehiclemodel varchar(40) NOT NULL,  
    vehiclenumber varchar(20) NOT NULL,  
    vehicleclass varchar(40) NOT NULL,  
    vehicletype varchar(30) NOT NULL,  
    userid number(11) NOT NULL,  
    CONSTRAINT vehiclenumber_unique UNIQUE (vehiclenumber)  
);  
  
CREATE SEQUENCE seq_vehicle MINVALUE 1 START WITH 1 INCREMENT BY 1  
CACHE 10;  
  
ALTER TABLE vehicle ADD FOREIGN KEY (userid ) REFERENCES users(userid );
```







8.4. TRANSACTION:

```
CREATE TABLE transaction (  
    tid number(11) NOT NULL PRIMARY KEY,  
    placeName varchar(30) NOT NULL,  
    amount varchar(30) NOT NULL,  
    time varchar(30) NOT NULL,  
    userid varchar(30) NOT NULL,  
    vehiclenumber varchar(30) NOT NULL,  
    VEHICLEID varchar(30) NOT NULL );  
  
CREATE SEQUENCE seq_transaction MINVALUE 1 START WITH 1 INCREMENT  
BY 1 CACHE 10;
```







ALTER TABLE transaction ADD FOREIGN KEY (vehicleid) REFERENCES vehicle(vehicleid);

9. TABLE DATA:

9.1. USERS:

| EDIT | USERID | USERNAME | PASSWORD | FULLNAME | PHONE | DOB | ADMIN |
|---|--------|----------|----------|-----------------|-------------|-----------------|-------|
|  | 12 | noor | 2 | ZUHAN NOOR | 01236458791 | 1 January, 2000 | 0 |
|  | 13 | rafi | 1236987 | RAFI AHMED | 01284445698 | 4 April, 1988 | 0 |
|  | 14 | rana | 147852 | MD. SOHEL RANA | 01456987561 | 7 April, 1996 | 0 |
|  | 15 | ripon | 15698745 | Ripon Khan | 01845698521 | 4 March, 1995 | 0 |
|  | 16 | roni | 156789 | RONI SHIKDAR | 01123645789 | 4 May, 1999 | 0 |
|  | 17 | shojib | 1 | SHOJIBUR RAHMAN | 01827056645 | 30 June, 1998 | 1 |
| row(s) 1 - 6 of 6 | | | | | | | |








9.2. VEHICLE:

| EDIT | VEHICLEID | BRANDNAME | VEHICLEMODEL | VEHICLENUMBER | VEHICLECLASS | VEHICLETYPE | USERID |
|---|-----------|-----------|--------------|-----------------|-----------------|-------------|--------|
|  | 1 | McLaren | F1 | AA-OOPS | Private service | car | 12 |
|  | 2 | Bajaj | Pulsar | Dhaka 11-9968 | Motorcycle | up to 50 cc | 13 |
|  | 12 | Bajaj | Pulsar | Dhaka 39-5569 | Motorcycle | up to 50 cc | 15 |
|  | 13 | Toyota | Allion | Khulna 78-36987 | Private service | car | 16 |
|  | 14 | Toyota | Premio | Sylhet 56-9784 | Private service | car | 13 |
|  | 3 | Toyota | Allion | Dhaka 12-6665 | Private service | car | 14 |
| row(s) 1 - 6 of 6 | | | | | | | |

9.3. TOLL:

| EDIT | TOLLID | PLACE | PRICE |
|---|--------|---------|-------|
|  | 1 | jamuna | 500 |
|  | 4 | meghna | 300 |
|  | 5 | maghna2 | 350 |
|  | 2 | padma | 600 |
|  | 3 | gomti | 200 |
| row(s) 1 - 5 of 5 | | | |

9.4. TRANSACTION

| EDIT | TID | PLACENAME | AMOUNT | TIME | VEHICLEID | VEHICLENUMBER |
|---|-----|-----------|--------|---------------------|-----------|-----------------|
|  | 1 | meghna | 500 | 2019/11/01 17:44:01 | 1 | Sylhet 56-9784 |
|  | 2 | gomti | 500 | 2019/11/01 17:44:25 | 2 | Khulna 78-36987 |
|  | 3 | meghna | 500 | 2019/11/01 17:44:46 | 3 | Dhaka 12-6665 |
|  | 4 | Kachpur | 250 | 2019/11/01 17:45:18 | 13 | Dhaka 39-5569 |
|  | 5 | padma | 450 | 2019/11/01 17:45:22 | 13 | Dhaka 39-5569 |
|  | 6 | jamuna | 560 | 2019/11/04 22:27:55 | 1 | Dhaka 11-9968 |
|  | 10 | gomti | 500 | 2019/11/04 22:27:59 | 1 | Sylhet 56-9784 |
| row(s) 1 - 7 of 7 | | | | | | |

10. VIEW

1. create view VEHICLE_view as

```
select  
VEHICLEID,BRANDNAME,VEHICLEMODEL,VEHICLENUMBER,VEHICLECLASS,VEH  
ICLETYPE,USERID  
from VEHICLE;
```

2. create view user_view as

```
select USERID,USERNAME,PASSWORD,FULLNAME,PHONE,DOB  
from USERS;
```

3. create view comb_user_vehicle as

```
select u.USERNAME,v.VEHICLENUMBER,v.BRANDNAME,v.VEHICLETYPE  
from users u,VEHICLE v  
where u.userid=v.userid;
```

4. create view comb_trans_vehicle as

```
select  
t.PLACENAME,t.AMOUNT,t.TIME,v.VEHICLETYPE,v.VEHICLEMODEL,v.VEHICLENU  
MBER  
from VEHICLE v,TRANSACTION t where v.VEHICLEID=t.VEHICLEID;
```

11. PROCEDURE

1. create or replace procedure p_getuserID(un IN USERS.USERNAME%type, ui OUT USERS.USERID%type) AS

BEGIN

ui := getuserID(un);

END;

2. create or replace procedure p_isadmin(un IN USERS.USERNAME%type, a OUT USERS.USERID%type) AS

BEGIN

a := isadmin(un);

END;

3. create or replace procedure p_getvehicleid(vn IN VEHICLE.VEHICLENUMBER%type, v OUT VEHICLE.VEHICLEID %type) AS

BEGIN

v := getvehicleid(vn);

END;

4. create or replace procedure p_add_vehicle(bn IN VEHICLE.BRANDNAME%type, vm IN VEHICLE.VEHICLEMODEL%type, vn IN VEHICLE.VEHICLENUMBER%type, vc IN VEHICLE.VEHICLECLASS%type,

vt IN VEHICLE.VEHICLETYPE%type, ui IN VEHICLE.USERID%type, a OUT VEHICLE.USERID%type) AS

BEGIN

a := add_vehicle(bn, vm, vn, vc, vt, ui);

END;

5. create or replace procedure p_delete_vehicle(vn IN VEHICLE.vehiclenumber%type, a OUT VEHICLE.USERID%type) AS

BEGIN

a := delete_vehicle(vn);

END;

6. create or replace procedure p_update_password(np IN USERS.PASSWORD%type, ui IN USERS.USERNAME%type, cp IN USERS.PASSWORD%type, a OUT USERS.USERID%type)

AS

BEGIN

a := update_password(np, ui, cp);

END;

12. FUNCTION

1. create or replace function getuserID(un USERS.USERNAME%type) return NUMBER

is

id NUMBER(11);

BEGIN

select userid into id from users where username = un;

return id;

END;

2. create or replace function isadmin(un USERS.USERNAME%type) return NUMBER

is

a NUMBER(11);

BEGIN

select admin into a from users where username = un;

return a;

END;

3. create or replace function getvehicleid(vn VEHICLE.VEHICLENUMBER%type) return NUMBER

is

v NUMBER(11);

BEGIN

select VEHICLEID into v from VEHICLE where VEHICLENUMBER = vn;

return v;

END;

4. create or replace function add_vehicle(bn VEHICLE.BRANDNAME%type, vm VEHICLE.VEHICLEMODEL%type, vn VEHICLE.VEHICLENUMBER%type, vc VEHICLE.VEHICLECLASS%type,

vt VEHICLE.VEHICLETYPE%type, ui VEHICLE.USERID%type) return number

is

BEGIN

INSERT INTO vehicle (vehicleid, brandname, vehiclemodel, vehiclenunder, vehicleclass, vehicle type, userid) VALUES (seq_vehicle.nextval, bn, vm, vn, vc, vt, ui);

return ui;

END;

5. create or replace function delete_vehicle(vn VEHICLE.vehiclenumber%type) return number
is

BEGIN

DELETE FROM vehicle WHERE vehiclenumber = vn;

return 1;

END;

6. create or replace function update_password(np USERS.PASSWORD%type, ui
USERS.USERNAME%type, cp USERS.PASSWORD%type) return number

is

BEGIN

UPDATE users SET password = np WHERE username = ui AND password = cp;

return 1;

END;

13. TRIGGER

1. CREATE or REPLACE TRIGGER office_time

BEFORE INSERT OR UPDATE OR DELETE ON TOLL

FOR EACH ROW

BEGIN

IF(TO_CHAR(SYSDATE,'DAY')in('FRI','SAT') or TO_CHAR(SYSDATE,'HH24') NOT BETWEEN '08' and '22') then

raise_application_error(-20501,'INVALID TIMING');

end if;

END;

2. CREATE or REPLACE TRIGGER toll_price_limit

BEFORE INSERT OR UPDATE ON TOLL

FOR EACH ROW

BEGIN

IF(:new.PRICE NOT BETWEEN 100 and 1000) then

raise_application_error(-20500,'INVALID AMOUNT');

end if;

END;

3. CREATE or REPLACE TRIGGER pay_price_limit

BEFORE INSERT ON TRANSACTION

FOR EACH ROW

BEGIN

IF(:new.AMOUNT < 0) then

raise_application_error(-20504,'INVALID AMOUNT');

end if;

END;

```
4. CREATE or REPLACE TRIGGER office_time_users_update
BEFORE DELETE ON users
FOR EACH ROW
BEGIN
IF(TO_CHAR(SYSDATE,'DAY')in('FRI','SAT') or TO_CHAR(SYSDATE,'HH24') NOT
BETWEEN '08' and '17') then
raise_application_error(-20507,'INVALID TIMING');
end if;
END;
```