

OnePoint Vehicle Management System

Transactions, Expenses and Business Management

Project By

Hassan Alvi	27121
Muhammad Saad Saghir	27143
Muhammad Ali	27142
Muhammad Iftikhar	27151

CONTENT PAGE

1. Abstract

2. Background

3. Introduction

- a. Introduction to the System
- b. Purpose of the project
- c. Problems in existing system
- d. Solutions to those problems

4. System Analysis

5. System Design

6. Normalization (Un-normalized and Normalization upto 3NF)

7. System Design (Tables, Views, Users, Triggers, Exception etc)

8. Desktop Application Forms

9. Conclusion

ABSTRACT

To develop a feature rich desktop solution for a typical transportation department of a particular manufacturing company in a typical software development environment.

The “one point Vehicle Management System” keeps track the information about the Vehicles, Maintenance, Repair, Parts, and Employees. It also keeps track the maintenance performed for different vehicles which are used for transportation.

The super users of the system are the ‘ADMIN’ and the ‘users’ are allocated by the admin. The admin may be the owner of the transportation organization or the user of transportation department of a particular manufacturing company.

If any other vehicle is added to the fleet which already exists for the organization/department the details of the vehicle is added. The details includes whether it is a new one or taken any loan/lease. Any employee is newly appointed or the existing employee is taken off both the details are maintained including their personal details and profession details.

The details of the maintenance that are being performed such as repairs/services and to be performed are also maintained. The maintenance to be performed can also be scheduled for each type of vehicles. The details of the parts/inventory used for the vehicles are maintained. The reorder level and the reorder quantity are predefined for each particular type of part.

The Vendors or suppliers of the vehicles, parts and performs maintenance required for the vehicles. The particulars of the various vendors are maintained in this system.

Existing System:

- Managing huge Fleet information manually is a tedious and error prone task.
- In order to schedule vehicles as well as staff, we the scheduler should not how many vehicles are there on board and available for allocation.
- Keeping track of repair information is a must as some times vehicles might be referred for insurance.
- This entire thing cannot be achieved in existing system.

Proposed System:

- In order to avoid the limitations in the existing system, the current system is being developed.
- All vehicle details will be automated along with the staff information.
- Scheduling of trips and repair information is being fully automated to overcome chaos in the system.

a. INTRODUCTION & OBJECTIVE

The ‘one point Vehicle Management System’ keeps track the information about the Vehicles, Maintenance, Repair, Parts, and Employees. It also keeps track the maintenance performed for different vehicles which are used for transportation.

INTRODUCTION

The Vehicle Management System (VMS) is an application for Transporters. It supports, in the area of Sales & Services, the business processes that you require as vehicle importer when dealing with your original equipment manufacturers (OEMs) and your dealers in new and used vehicle sales. VMS offers you complete integration of all the relevant processes such as procurement, sales, rework, returns processing, trade-in and service processing. It also supports the archiving of vehicle data. In other words, it allows you to react flexibly to customers' requirements in the area of production (using the "pull strategy") and fast delivery times with reduced warehouse stock and sales/distribution costs.

There are mainly 5 modules in this software

- Vehicle Management
- Driver Management
- User Management
- Transaction Management
- Expense Management

In vehicle module a user can add a new vehicle details to the database. Vehicle details like the company, model name, price and the availability of the vehicle are added. Whenever the user wants to modify these details he can update new values through this software. Driver Management module deals with the drivers management of the company. A user can add the information about the new driver and if he wants he can change it or delete that driver. The main advantage is that this module is helpful for the agents to get details of the drivers. User details can be added to the database and this is helpful as it manages all the users using this program. An operator can add the personal details and if we want to edit or delete he can do all modification using this system. Transaction management manages the orders we take from the users. It also gives a hand in renting of the cars from our system. We can add, edit and delete the details of the renting as per their requirements and provide the users with the desired cars, accordingly.

SYSTEM ANALYSIS

EXISTING SYSTEM

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

During analysis, data collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, interviews, etc. Training, experience and common sense are required for collection of relevant information needed to develop the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

System analysis can be categorized into four parts.

- ✓ System planning and initial investigation
- ✓ Information Gathering
- ✓ Applying analysis tools for structured analysis

- ✓ Feasibility study
- ✓ Cost/ Benefit analysis

In the existing system all the jobs of the vehicle management is done manually. This is very difficult to the operators who want to handle hundreds of customers and many buses in a day. The current system is that an operator wants to keep the physical records of the vehicles in his office and a separate record for the customers. In current system there is no way to store the details of the employees' working in the system. So many complaints against staff can arise from the customers' side. More over there is no detailed record of the vehicles and customers.

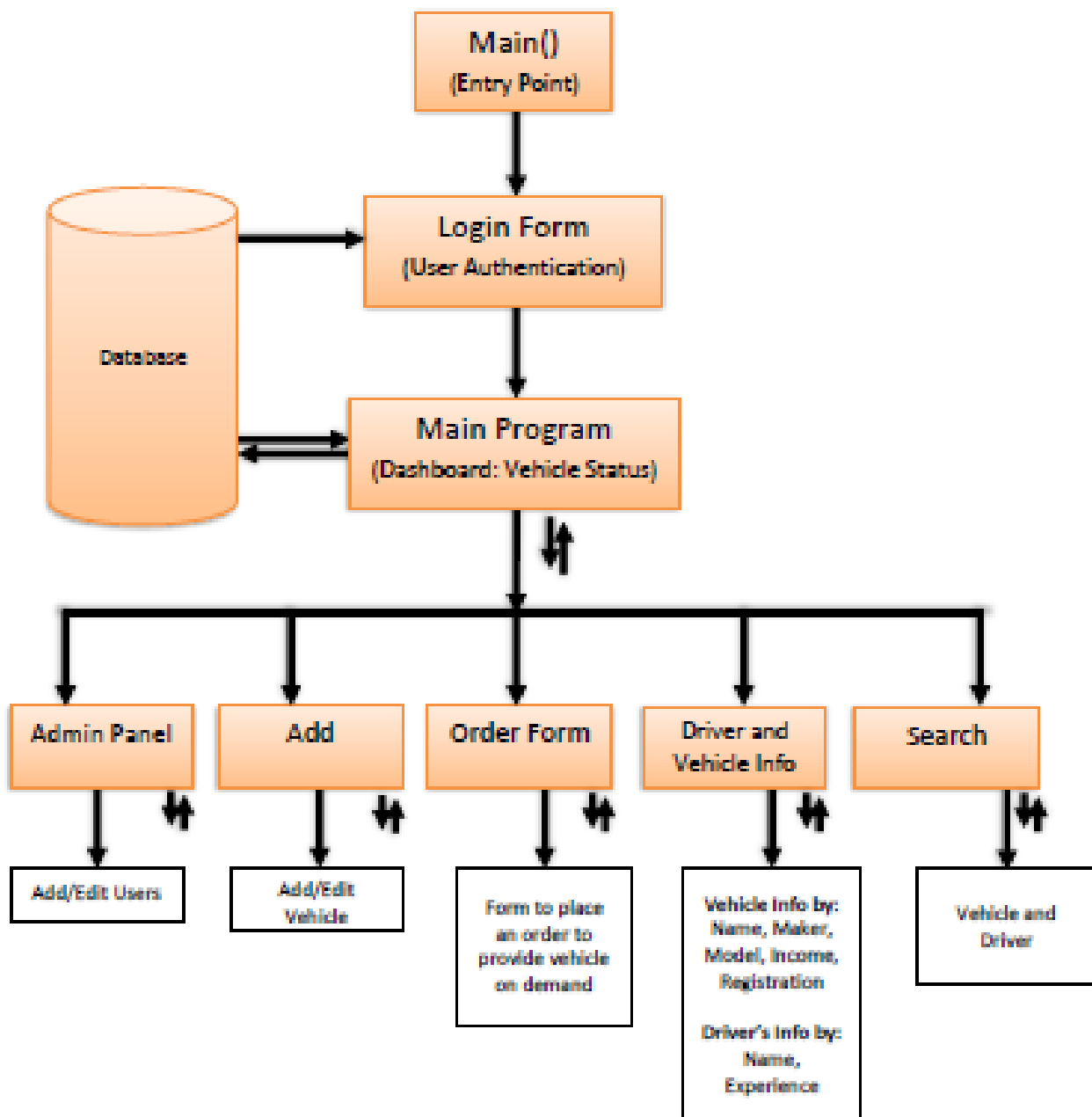
PROPOSED SYSTEM

The proposed system is very useful for the operators and customers. This avoids the overheads for the operators. They can minimize the working stress and can keep essential documents related to the vehicles and the customers as a softcopy. The advantage of the proposed system is the reduction in the cost of the office equipment and the transaction is done quickly. Any operator can answer if any vehicle is available in a particular day or not.

Our proposed system has several advantages

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Look and Feel Environment
- Quick transaction

STRUCTURE CHART



FEASIBILITY ANALYSIS

Whatever we think need not be feasible .It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

TECHNICAL FEASIBILITY

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

ECONOMIC FEASIBILITY

Development of this application is highly economically feasible .The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. I \f we are doing so , we can attain the maximum usability of the corresponding resources .Even after the development , the organization will not be in a condition to invest more in the organization .Therefore , the system is economically feasibility.

SYSTEM DESIGN

INPUT DESIGN

Input design is the process of converting user-oriented input to a computer based format. Input design is a part of overall system design, which requires very careful attention .Often the collection of input data is the most expensive part of the system. The main objectives of the input design are ...

- Produce cost effective method of input
- Achieve highest possible level of accuracy
- Ensure that the input is acceptable to and understood by the staff.

INPUT DATA

The goal of designing input data is to make entry easy, logical and free from errors as possible. Input stages include the following processes

- Data Recording
- Data Transcription
- Data Conversion
- Data Verification
- Data Control
- Data Transmission
- Data Correction

One of the aims of the system analyst must be to select data capture method and devices, which reduce the number of stages so as to reduce both the changes of errors and the cost.

Input types, can be characterized as.

- External
- Internal
- Operational
- Computerized
- Interactive

Input files can exist in document form before being input to the computer. Input design is rather complex since it involves procedures for capturing data as well as inputting it to the computer.

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these result for latter consultation .Computer output is the most important and direct source of information to the users. Designing computer output should proceed in an organized well throughout the manner. The right output must be available for the people who find the system easy o use. The outputs have been defined during the logical design stage. If not, they should defined at the beginning of the output designing terms of types of output connect, format, response etc. Various types of outputs are:

- External outputs
- Internal outputs
- Operational outputs
- Interactive outputs
- Turn around outputs

All screens are informative and interactive in such a way that the user can fulfill his requirements through asking queries.

DATABASE DESIGN

- The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users.

Normalization Process

Un-Normalized Data															
VehicleID	Maker	Model	Type	Made	Engine	Chasis	Size	Price	PurchaseDate	FitnessValid	FitnessExpiry	Status			
DriverID	driverSince	Name	FatherName	Age	CNIC	Mobile	Experience	Accidents	Salary	Availability					
MaintenanceID	partRepaired	date	Description	PartPrice	RpicDate										
TransactionID	transDate	clientName	CNIC	mobile	tripType	Source	Current_Loc	Destination	kilometers	Charges	LateCharges	Status			
1NF															
PK	PK	PK													
TransactionID	vehicleID	DriverID	maintenanceID	Maker	Model	Type	Made	Engine	Chasis	Size	Price	PurchaseDate	FitnessValid	FitnessExpiry	Status
partRepaired	date	Description	PartPrice	RpicDate	transDate	clientName	CNIC	mobile	tripType	Source	Current_Loc	Destination	kilometers	Charges	
LateCharges	Status	expenseDate	description	amount	userID	Type	event	user	logOnDate	logOnTime	logOffDate	logOffTime			

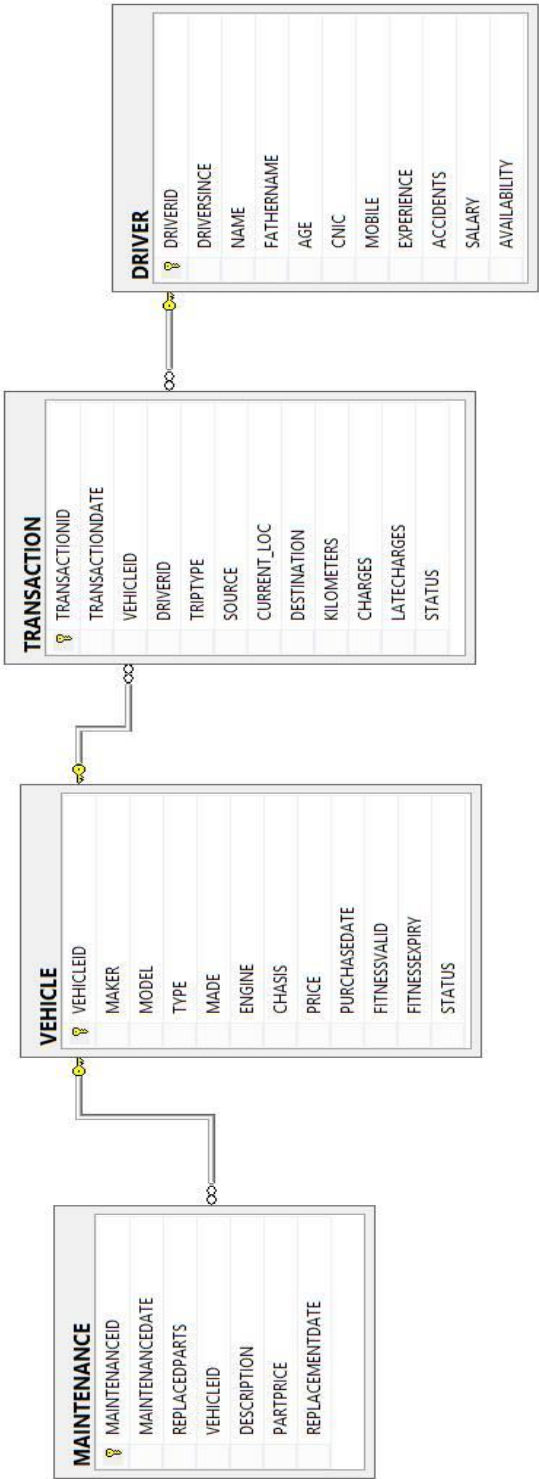
Normalization Process (Contd...)

2NF												
Vehicle												
PK												
VehicleID	Maker	Model	Type	Made	Engine	Chasis	Size	Price	PurchaseDate	FitnessValid	FitnessExpiry	Status
varchar	varchar	varchar	varchar	DATE	varchar	varchar	varchar	money	DATE	Date	Date	varchar
Driver												
PK												
DriverID	driveSince	Name	FatherName	Age	CNIC	Mobile	Experience	Accidents	Salary	Availability		
varchar	Date	varchar	varchar	number	varchar	varchar	number	number	number	varchar		
Maintenance												
PK												
MaintenanceID	VehicleID	partRepalced	date	Description	PartPrice	ReplacementDate						
varchar	varchar	varchar	Date	varchar	money	date						
Transaction												
PK					FK	FK						
TransactionID	transDate	clientName	CNIC	mobile	VehicleID	DriverID	tripType	Source	Current_Loc	Destination	kilometers	Status
varchar	DATE	varchar	varchar	varchar	varchar	varchar	char	varchar	varchar	varchar	number	varchar
expenseDate	description	amount	userID	Type	event	user	logOnDate	logOnTime	logOffDate	logOffTime		
date	varchar	number	varchar	varchar	varchar	varchar	date	date/time	date	date/time		

Normalization Process (Contd...)

3NF														
PK	Vehicle													
	VehicleID	Maker	Model	Type	Made	Engine	Chassis	Size	Price	PurchaseDate	FitnessValid	FitnessExpiry	Status	
	varchar	varchar	varchar	varchar	DATE	varchar	varchar	varchar	money	DATE	Date	Date	varchar	
PK	Driver													
	DriverID	driveSince	Name	FatherName	Age	CNIC	Mobile	Experience	Accidents	Salary	Availability			
	varchar	Date	varchar	varchar	number	varchar	varchar	number	number	number	varchar			
PK	Maintenance													
	MaintenanceID	VehicleID	partRepalced	date	Description	PartPrice	ReplacementDate							
	varchar	varchar	varchar	Date	varchar	money	date							
PK	Transaction													
	TransactionID	transDate	clientName	CNIC	mobile	VehicleID	DriverID	tripType	Source	Current_Loc	Destination	kilometers	Charges	LateCharges
	varchar	DATE	varchar	varchar	varchar	varchar	varchar	char	varchar	varchar	varchar	number	money	money
Expenses	Users													
	expenseDate	description	amount	userID	userID	Type		event	user	logOnDate	logOnTime	logOffDate	logOffTime	
	date	varchar	number	varchar	varchar	varchar		varchar	varchar	date	date/time	date	date/time	

Entity Relationship Diagram



Database Design

```
SQL Plus

SQL> create user DBADMIN identified by dbAdmin account unlock;
User created.

SQL> grant all privileges, dba to dbadmin;
Grant succeeded.

SQL>
SQL>
SQL> _
```

```
SQL Plus

SQL> create or replace function TransactionsToday
2 return number
3 is
4     totalTrans number;
5 begin
6     select count(*) into totalTrans from dbadmin.transaction where transactiondate = trunc(sysdate);
7     if (totalTrans = 0) then
8         raise_application_error(-20001, 'No Record Found!');
9     end if;
10    return totalTrans;
11 end;
12 /
Function created.
SQL>
```

```
SQL Plus

SQL> CREATE OR REPLACE FUNCTION TotalExpenses(dateFrom in Date, dateTo in Date)
2 return number
3 is
4     totalAmount number;
5 begin
6     select sum(amount) into totalAmount from dbadmin.expenses where expensedate between dateFrom and dateTo;
7     if (totalAmount = 0) then
8         raise_application_error(-20001, 'No Record Found!');
9     else
10        return totalAmount;
11    end if;
12 end;
13 /
Function created.
SQL> _
```

```
SQL> grant execute on transactionsToday to user123;
Grant succeeded.
SQL> grant execute on totalExpenses to user123;
Grant succeeded.
SQL> _
```

```
SQL> create or replace procedure DELETEDRIVER(driver_ID in varchar2)
2 is
3     drvID varchar2(10);
4 begin
5     select driverID into drvID from dbadmin.driver where driverID = driver_ID;
6     delete from dbadmin.driver where driverID = drvID;
7     commit;
8 exception
9     when no_data_found then
10         raise_application_error(-20001, 'No Data Found!');
11 end;
12 /
Procedure created.
SQL> _
```

```
SQL> create or replace
2 procedure deleteUser(user_ID in varchar2)
3 is
4     usrID varchar2(10);
5 begin
6     select userID into usrID from dbadmin.users where userID = user_ID;
7     delete from dbadmin.users where userID = usrID;
8     commit;
9 exception
10     when no_data_found then
11         raise_application_error(-20001, 'No Data Found!');
12 end;
13 /
Procedure created.
SQL> execute deleteUser('USR-101');
BEGIN deleteUser('USR-101'); END;
*
ERROR at line 1:
ORA-20001: No Data Found!
ORA-06512: at "DBADMIN.DELETEUSER", line 10
ORA-06512: at line 1
SQL> _
```

```
SQL Plus

SQL> create or replace
2 procedure deleteVehicle(vehicle_ID in varchar2)
3 is
4   vhcID varchar2(10);
5 begin
6   select vehicleID into vhcID from dbadmin.vehicle where vehicleID = vehicle_ID;
7   delete from dbadmin.vehicle where vehicleID = vhcID;
8   commit;
9 exception
10  when no_data_found then
11    raise_application_error(-20001, 'No Data Found!');
12 end;
13 /

Procedure created.

SQL>
```

```
SQL> show user;
USER is "DBADMIN"
SQL>
SQL>
SQL> CREATE TABLE DRIVER
2 <
3   DRIVERID VARCHAR2(10),
4   DRIVERSINCE DATE,
5   NAME VARCHAR2(30),
6   FATHERNAME VARCHAR2(30),
7   AGE NUMBER(2),
8   CNIC VARCHAR2(15),
9   MOBILE VARCHAR2(15),
10  EXPERIENCE NUMBER(2),
11  ACCIDENTS NUMBER(2),
12  SALARY NUMBER(5),
13  AVAILABILITY CHARACTER,
14  CONSTRAINTS DRURID_PK PRIMARY KEY (DRIVERID)
15 >;

Table created.

SQL> _
```

```
SQL> show user;
USER is "DBADMIN"
SQL>
SQL>
SQL>
SQL> CREATE TABLE EVENT_LOG
2 <
3   EVENT VARCHAR2(10),
4   USERNAME VARCHAR2(15),
5   LOGON_DATE DATE,
6   LOGON_TIME VARCHAR2(18),
7   LOGOF_DATE DATE,
8   LOGOF_TIME VARCHAR2(18)
9 >;

Table created.

SQL>
SQL>
SQL> _
```

```
SQL Plus

SQL> SHOW USER;
USER is "DBADMIN"
SQL>
SQL>
SQL> CREATE TABLE MAINTENANCE
 2  (
 3    MAINTENANCEID VARCHAR2(5),
 4    MAINTENANCEDATE DATE,
 5    REPLACEDPARTS VARCHAR2(20),
 6    VEHICLEID VARCHAR2(20),
 7    DESCRIPTION VARCHAR2(50),
 8    PARTPRICE NUMBER(6),
 9    REPLACEMENTDATE DATE,
10    CONSTRAINTS MNINCID_PK PRIMARY KEY (MAINTENANCEID),
11    CONSTRAINTS UHCID_PFK FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID)
12  );

Table created.

SQL>
SQL> _
```

```
SQL Plus

SQL> SHOW USER;
USER is "DBADMIN"
SQL>
SQL>
SQL> CREATE TABLE TRANSACTION
 2  (
 3    TRANSACTIONID VARCHAR2(10),
 4    TRANSACTIONDATE DATE,
 5    VEHICLEID VARCHAR2(10),
 6    DRIVERID VARCHAR2(10),
 7    TRIPTYPE CHARACTER,
 8    SOURCE VARCHAR2(30),
 9    CURRENT_LOC VARCHAR2(30),
10    DESTINATION VARCHAR2(30),
11    KILOMETERS NUMBER(4),
12    CHARGES NUMBER(6),
13    LATECHARGES NUMBER(6),
14    STATUS VARCHAR2(8),
15    CONSTRAINTS TRNSTNID_PK PRIMARY KEY (TRANSACTIONID),
16    CONSTRAINTS UHCID_TFK FOREIGN KEY (VEHICLEID) REFERENCES VEHICLE(VEHICLEID),
17    CONSTRAINTS DRUID_TFK FOREIGN KEY (DRIVERID) REFERENCES DRIVER(DRIVERID)
18  );

Table created.

SQL>
```



```
SQL> show user;
USER is "DBADMIN"
SQL>
SQL>
SQL> CREATE TABLE TransactionsLog AS
2 SELECT * FROM TRANSACTION WHERE 1 = 2;

Table created.
SQL>
```

```
SQL> CREATE TABLE VEHICLE
2 <
3     VEHICLEID VARCHAR2(10),
4     MAKER VARCHAR2(10),
5     MODEL VARCHAR2(10),
6     TYPE VARCHAR2(10),
7     MADE DATE,
8     ENGINE VARCHAR2(10),
9     CHASIS VARCHAR2(10),
10    PRICE NUMBER(8),
11    PURCHASEDATE DATE,
12    FITNESSVALID DATE,
13    FITNESSEXPIRY DATE,
14    STATUS VARCHAR2(5),
15    CONSTRAINTS UHCID_PK PRIMARY KEY <VEHICLEID>
16 >;
17

Table created.
SQL> SHOW USER;
USER is "DBADMIN"
SQL> _
```

```
SQL Plus

SQL> show user;
USER is "DBADMIN"
SQL>
SQL>
SQL> CREATE OR REPLACE TRIGGER LOGOFF_TRIGGER
2 BEFORE LOGOFF ON DATABASE
3 BEGIN
4     UPDATE EVENT_LOG
5     SET LOGOF_DATE = SYSDATE WHERE LOGOF_DATE IS NULL;
6
7     UPDATE EVENT_LOG
8     SET LOGOF_TIME = TO_CHAR(SYSDATE, 'hh24:mm:ss') WHERE LOGOF_TIME IS NULL;
9
10    COMMIT;
11 END;
12 /

Trigger created.
SQL>
```

```
SQL Plus
SQL> show user;
USER is "DBADMIN"
SQL>
SQL>
SQL> CREATE TABLE EVENT_LOG
2  (
3      EVENT VARCHAR2(10),
4      USERNAME VARCHAR2(15),
5      LOGON_DATE DATE,
6      LOGON_TIME VARCHAR2(18),
7      LOGOFF_DATE DATE,
8      LOGOFF_TIME VARCHAR2(18)
9  );

Table created.

SQL> CREATE OR REPLACE TRIGGER LOGON_TRIGGER
2  AFTER LOGON ON DATABASE
3  BEGIN
4      INSERT INTO EVENT_LOG
5      VALUES(
6          ORA_SYSEVENT,
7          USER,
8          SYSDATE,
9          TO_CHAR(SYSDATE, 'hh24:mm:ss'),
10         NULL,
11         NULL
12     );
13     COMMIT;
14 END;
15 /

Trigger created.

SQL>
```

```
SQL Plus
SQL> CREATE OR REPLACE TRIGGER TTRANS_UPDATE
2  AFTER UPDATE ON TRANSACTION
3  FOR EACH ROW
4  BEGIN
5      INSERT INTO TransactionsLog
6      SELECT * FROM TRANSACTION WHERE TRANSACTIONID = :NEW.TRANSACTIONID;
7  END;
8  /

Trigger created.

SQL> █
```

```

SQL> create or replace view UUDRIVERSLIST
2 as
3 select driverID "Driver ID", driverSince "Join Date", Name, Experience, Salary from dbadmin.driver;
View created.
SQL> select * from vudriverslist;
Driver ID  Join Date  NAME                                EXPERIENCE  SALARY
-----
DRV-001    01-APR-16  MUSHTAQ ALI                        5           18000
DRV-002    05-APR-16  ZUBAIR SHAH                        2           18000
DRV-003    08-APR-16  SALMAN AHMAD                       0           18000
SQL> _

```

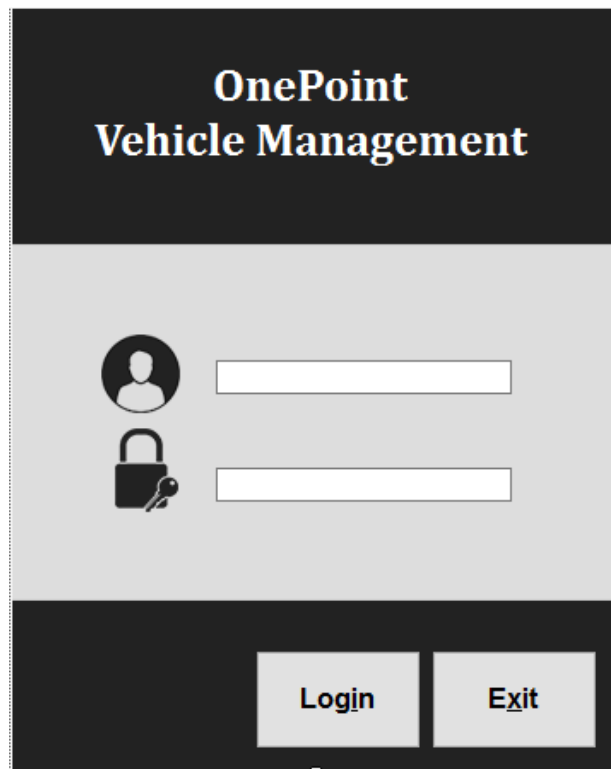
```

SQL> create view UUTransactionDone
2 as
3 select d.driverid, d.name, v.vehicleid, v.model, t.source, t.destination, t.kilometers
4 from dbadmin.transaction t join dbadmin.driver d on d.driverid = t.driverid
5 join dbadmin.vehicle v on v.vehicleid = t.vehicleid
6 where t.status = 'DONE';
View created.
SQL>
SQL>
SQL>
SQL> create view UUTransactionPending
2 as
3 select d.driverid, d.name, v.vehicleid, v.model, t.source, t.destination, t.kilometers
4 from dbadmin.transaction t join dbadmin.driver d on d.driverid = t.driverid
5 join dbadmin.vehicle v on v.vehicleid = t.vehicleid
6 where t.status = 'PENDING';
View created.
SQL>

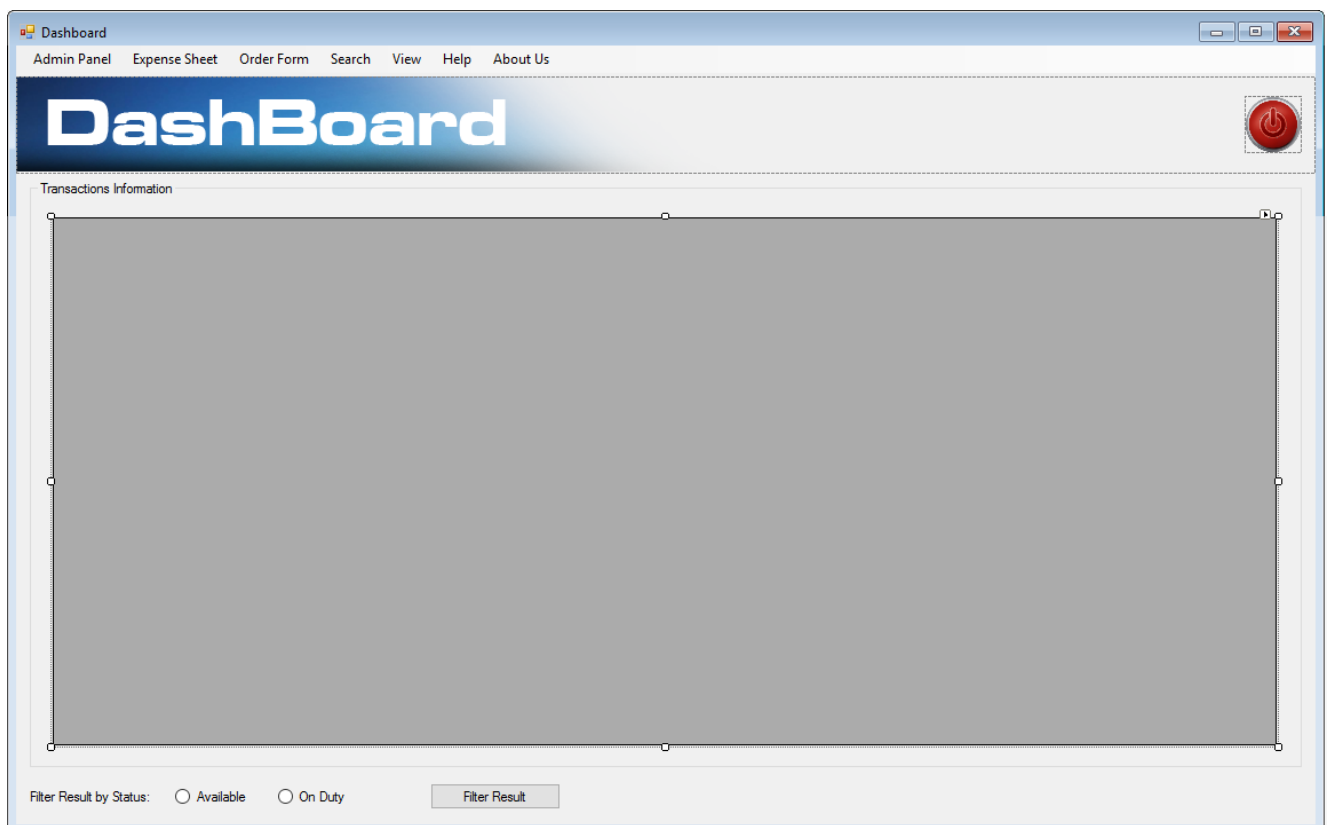
```

These are Objects that are currently in working state. We are working on other objects as well as the project makes progress (i.e. GUI and Oracle Objects are in progress concurrently).

Graphical User Interface of OnePoint VMS



The login screen for the OnePoint Vehicle Management system. It features a dark blue header with the text "OnePoint Vehicle Management" in white. Below the header is a light gray area containing two input fields. The first input field is preceded by a user icon, and the second input field is preceded by a padlock icon. At the bottom of the screen, there are two buttons: "Login" and "Exit", both in white text on a dark blue background.



The dashboard interface of the OnePoint Vehicle Management system. It has a blue header with the word "Dashboard" in white. Below the header is a menu bar with links: "Admin Panel", "Expense Sheet", "Order Form", "Search", "View", "Help", and "About Us". The main content area is titled "Transactions Information" and contains a large gray rectangular area, likely a placeholder for a table or chart. At the bottom, there is a filter section with the text "Filter Result by Status:" followed by two radio buttons labeled "Available" and "On Duty", and a "Filter Result" button.

Users

User Info

☐ Show Password

New User ID:

Password:

User Type: ☐ ADMIN ☐ USER

Add Update Delete

Refresh Clear All Exit

Users

Vehicle

VEHICLE FORM

Vehicle Info

Enter Vehicle ID: OR

Maker: Model:

Type: Made In (Year):

Engine Number: Chassis Number:

Purchasing (Amount): Purchase Date:

Fitness Certificate:

From: Expiry:


Availability Status: ☒ Available ☐ Unavailable / To be Delivered

Add New Update Delete Clear All Exit

Driver


Driver's Information

Driver ID (Unique):

Driver Since: 

Name:

Father:

Age: Mobile: 

CNIC: Driving Experience (Years):

Salary: Number of Accidents (in History):


Is Available? ☒ Yes ☐ No Now

Modification Section

Driver ID: OR

Drivers List

DRIVERS' LIST



Selected ID:

Search Driver

SEARCH DRIVER

Search Criteria: ☒ Unique ID ☐ Name ☐ Salary

Search Value: Enter ID: ☐ Exact ☐ Similar ☐ Not Any

Result

Search Refresh Cancel

Search Vehicle

SEARCH VEHICLE

Search Criteria: ☒ Unique ID ☐ Model ☐ Engine

Search Value: Enter ID: ☐ Exact ☐ Similar ☐ Not Any

Result

Search Refresh Cancel

Expenses Form

DAILY EXPENSES

Expenses

Date: 5/20/2016 Amount: _____

Description: _____

Add Entry Clear All Fields Exit

View Expense History

Expenses Detail

Date Range

☒ Daily (Today) ☐ Last 7 days ☐ Custom

From: 5/20/2016 To: 5/20/2016 View

Output

Selected Range Total: _____ Close

CONCLUSION

Our project is only a humble venture to satisfy the needs of a customer. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the system.

The objective of software planning is to provide a frame work that enables the operator to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. Last but not least it is not the work that played the ways to success but the **ALMIGHTY**.