TOUR MANAGEMENT SYSTEM

Project Report Submitted To

Mahatma Gandhi University,

Kottayam

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF DEGREE OF

Bachelor of Computer Application (BCA)

By

Abin Abraham (Reg: 160021113572)

Albert Gigi (Reg: 160021113576)

Sachin Geo Jacob (Reg: 160021113627)

Tony Scaria (Reg: 160021113637)

Under Supervision of

Asst.Prof. AMBILY MERLIN KURUVILA (MCA)

Department of Computer Applications

SAINTGITS COLLEGE OF APPLIED SCIENCES, PATHAMUTTOM

2018-2019

Saintgits College of Applied Sciences

Pathamuttom, Kottayam-Pin: 686532, Kerala

Ph-0481-2436169,2436170,

website : www.saintgits.org

DECLARATION

We **Abin Abraham, Albert Gigi , Sachin Geo Jacob and Tony Scaria,** hereby declare that this project titled **"TOUR MANAGEMENT SYSTEM''** is the original work done by us, under the guidance and support of **Asst.Prof. Ambily Merlin Kuruvila**, during the academic year 2018-2019.

We also declare that this report has been submitted by us fully or partially for the award of degree before. Further, this is submitted on the partial fulfilment of the award of the degree of Bachelor of Computer Application of Mahatma Gandhi University, Kottayam, Kerala.

Abin Abraham

Albert Gigi

Sachin Geo Jacob

Tony Scaria

Place: Pathamuttom

Date:

Saintgits College of Applied Sciences

Pathamuttom, Kottayam-Pin: 686532, Kerala

Ph-0481-2436169,2436170,

website : www.saintgits.org

CERTIFICATE

This is to certify that the project report entitled **"TOUR MANAGETEMENT SYSTEM"** is a bonafide report of the project work undertaken by **Abin Abraham (**Reg: 160021113572), **Albert Gigi** (Reg: 160021113576), **Sachin Geo Jacob** (Reg: 160021113627), **Tony Scaria** (Reg: 160021113637), fifth semester BCA student under my supervision and guidance, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Computer Application (BCA)** of **MAHATMA GANDHI UNIVERSITY, Kottayam Kerala.**

Asst.Prof. Ambily Merlin Kuruvila **Asst.Prof. Ambily Merlin Kuruvila**

Head of the Department **Project Guide**

Submitted for the viva-voice examination held on …….

Date: **External Examiner**

Saintgits College of Applied Sciences

Pathamuttom, Kottayam-Pin: 686532, Kerala

Ph-0481-2436169,2436170,

website : www.saintgits.org

ACKNOWLEDGEMENT

First and foremost, I would like to express my profound gratitude to **God Almighty** for his immense blessings upon me that led to the successful completion of this project

I thankfully acknowledge our Principal **Prof. M.C. Joseph** for giving me an opportunity to present this mini project. I express my heartfelt gratitude to my project guide, **Asst.Prof. Ambily Merlin Kuruvila,** HOD, Department of Computer Application, for her valuable guidance, support and encouragement during course of the project and in the preparation of the report. I have greatly benefited from his experience and knowledge. The help extended by all other staff numbers of the department are gratitude. I also remember with thanks to all my friends and well-wishers for their encouragement and support.

Abin Abraham (Reg: 160021113572)

Albert Gigi (Reg: 160021113576)

Sachin Geo Jacob (Reg: 160021113627)

Tony Scaria (Reg: 160021113637)

TABLE OF CONTENTS

INTRODUCTION

INTRODUCTION

1.1 PROJECT ABSTRACT

This Project on "Tour Management" is an offline stand-alone project. This project on Tour Management is an application designed to deal with the packages and the booking in a tour managing centre.

The purpose of this project is to develop a system for Tour management office. Admin creates his own account and manage the account by doing necessary updates, deletion and retrieval. This feature includes employee management, customer, vehicle and report.

Admin provides an account for employee and in that account he can manage the details of customer, vehicle, booking.

We have the following modules in our project:

- **ADMINISTRATION MODULE**
- **EMPLOYEE MODULE**

1.2 OBJECTIVE AND SCOPE

Clearly the scope of the project is to provide the solutions for Tour management office for organizing data using software applications. This software application helps administrator to update data in step by step process for booking packages for customer. Using this system management team can update customer's information, package information, booking details etc.....

1.2.1 MODULES

The entire working of this software **is** divided into different modules. They are:

Administrator Module

1. Employee Management: The records of employees working are managed here
2. Salary Management: Payment of salary of employee are manage here.
3. Packages: The records of different packages are displayed here.
4. Vehicle: Details regarding the associated vehicles are maintained here.
5. Customer: The records of customers are displayed here.
6. Booking : The records of booking are displayed here.
7. Reports : Reports like employee, customer and booking are done
8. Notification: Various notifications regarding meetings are added and manipulated in this module
9. Manage accounts : Options for managing accounts of users

Employee Modules

1. Packages: The records of different packages are displayed here.
2. Vehicle: Details regarding the associated vehicles are maintained here.
3. Customer: The records of customers are displayed here.
4. Booking : The records of booking are displayed here.
5. Reports : Reports like employee, customer and booking are done
6. Notification: Various notifications regarding meetings are added and manipulated in this module
7. Manage accounts : Options for managing accounts of users

1.3 PROBLEM STATEMENT AND PROJECT RELEVANCE

1.3.1 PROBLEM DEFINITION

The tour management system has to handles records of large number of packages, booking, maintenance details and the maintenance of these records of was difficult .it is very easy to handle we get an overall idea of the system. It is used for all manual.

1.3.2 EXISTING SYSTEM

In this system it is very difficult task to manage the records of each and it will take a lot of time. But it also increases the chance of errors. sometimes even after repeated cross checks errors are found which lead to wrong entry of details .it creates a problem when you need

details of any particular project. All these problems lead to rise of an alternative option. It was following

Disadvantages:

1) **Wastage of time:** As the computations are done manually, takes lot of time to complete the works
2) **Inaccuracy:** As the computations are done manually, the chances for occurrence of errors is very high
3) **Data checking is very difficult:** If we want to check the data, we have to search from the beginning until we find it so the data checking also take lot of time**.**
4) **Paper file updating and keeping is a tedious process**: Data are stored to some extend on paper file. updating of file requires manual correction, which requires the physical transfer of file from one place to another. This will consume more time.
5) **Loss of corruption of records:** There are chances for loss of records due to fire or some other reasons such as changes made by other persons
6) **Human labour: When** processing is done manually, more manpower is needed

1.3.3 Proposed System

The software is very easy to use and compatible with the working environment. The work load is reduced and the entire details needed can be accessed very easily and at a faster rate. The features of the software help in placing the records safe and secured and the information regarding any case can be retrieved as and when required at a single click. The adding of new records, updating or deleting of existing files can be done more easily. It reduces wastage of time in recording of details and retrieval of information and also reduces errors that are most possible to occur. The records are organized in a standard way.

In short, by developing the system we attain the following features;

- Security of data.
- Ensured data accuracy's.
- Proper control of the higher officials.
- Reduce the damages of machines.

- Minimise manual data entry.
- Greater efficiency.
- Better service.
- Minimum time required.
- User friendliness and interactive.

SYSTEM ANALYSIS AND REQUIREMENT SPECIFICATION

2.1 REQUIREMENT STUDY

2.1.1 Feasibility report

Feasibility is to determine whether the proposed system is technically, economically and behaviourally feasible in all respect. The main aim of feasibility study is to evaluate alternative system and proposed the most feasible and desirable system for development. If there is no loss for the organization, then the proposed system is considered financially feasible.

- The project gives the instant responses to all sorts of queries that are being raised.
- The resources included are capable and can hold all types of data.
- The project will work on any platform if it is packaged and deployed.
- The project can be further expanded to satisfy the needs of the company.

HARDWARE REQUIREMENTS

CPU	: P111 and above processor
HARD DISK SPACE	: 40GB and above
DISPLAY	: 15'' MONITOR
MEMORY	: 256 and above
OTHER DEVICES	: DVD Drive

SOFTWARE REQUIREMENTS

OPERATING SYSTEM: Window 10 or compatible

FRONT- END: Visual Basic

BACK-END: Microsoft SQL server 2008

2.1.2 Economic Feasibility

This involves questions such as heather the company can afford to build the system, whether its benefits should substantially exceed its cost, and whether the project has higher priority and profits than the existing system. Here there is no problem. The company had a well-equipped hardware and software, so no need to spend money on these issues. And as the client and developer are one, there is no further problem in economic issues.

2.2 Requirement Analysis and Specification

Requirement analysis involves studying the current system to find out how it works and where improvements could be made clear idea about the existing system is a must for making improvements where it requires. Proper planning and collection of data services the purpose. The system manages all the information about a tour managing center.

2.2.1 System Requirements

System Requirements specification is a collection of information that incorporates the requirements of a system. This gives an idea about the system specification required to develop and install the project ‘’**TOUR MANAGEMENT SYSTEM’’**. The software Requirement specification is a technical specification of requirements for the software products. The goal of software requirements definition is to completely and consistently specify the technical requirements for the software products in a concise and unambiguous manner. The software Requirement specification is based on the system Definition. The requirement specifications are primarily concerned with functional and performance aspects of a software product and emphasis are placed on specifying product characteristics implying how his product will provide those characteristics.

One of the difficult tasks is selecting software, once the system requirement is found out then we have to determine whether a particular software package fits for those system requirements. This section summarizes the application requirements.

2.2.2 Hardware Requirements

CPU: P111 and above processor

HARD DISK SPACE: 40GB and above

RAM: 256 and above

DISPLAY: 15'' MONITOR

2.2.3 Software Requirement

OPERATING SYSTEM: Microsoft Windows 8

FRONT -END: Visual Basic 6.0

BACK –END: Microsoft SQL Server 2008

SYSTEM DESIGN&DEVELOPMENT

3.1 MODULES

The entire working of this software is divided into different modules. They are:

Administrator Modules

1. Employee Management: The records of employees working are managed here
2. Salary Management: Payment of salary of employee are manage here.
3. Packages: The records of different packages are displayed here.
4. Vehicle: Details regarding the associated vehicles are maintained here.
5. Customer: The records of customers are displayed here.
6. Booking : The records of booking are displayed here.
7. Reports : Reports like employee, customer and booking are done
8. Notification: Various notifications regarding meetings are added and manipulated in this module
9. Manage accounts : Options for managing accounts of users

Employee Modules

1. Packages: The records of different packages are displayed here.
2. Vehicle: Details regarding the associated vehicles are maintained here.
3. Customer: The records of customers are displayed here.
4. Booking : The records of booking are displayed here.
5. Reports : Reports like employee, customer and booking are done
6. Notification: Various notifications regarding meetings are added and manipulated in this module
7. Manage accounts : Options for managing accounts of users

3.2 DATA FLOW DIAGRAM

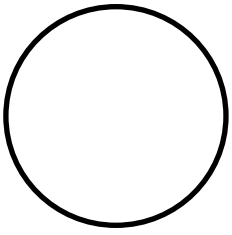
A data flow diagram (DFD) is a network that describes the flow of data throughout a system, data stores and processes that change, or transform, data flows. The DFD network is formal, logical abstract of a system that may have many possible physical configurations. For this reason, set of symbols that do not imply a physical form is used to request data sources, data flows, data transformation and data system storage they represent one of the most ingenious tools used for structured analysis. It has the purpose of clarifying system design. It is the major starting pointing the design phase the functionality decomposes the requirement specification down to the lowest level of detail. Structured analysis follows a top –down approach to describe the system. An initial model of the system only shows major activities of the system and this model is then explored into more detailed lower level diagrams. Each process in the system can be described with data flow diagram and can be broken into several another process to fully understand the system. In normal convention, the DFD has four major symbols.

Symbols

The following are the symbols used in the Data Flow Diagram

: **Square**, this defines source or destination of data

: Arrow, which shows data flow

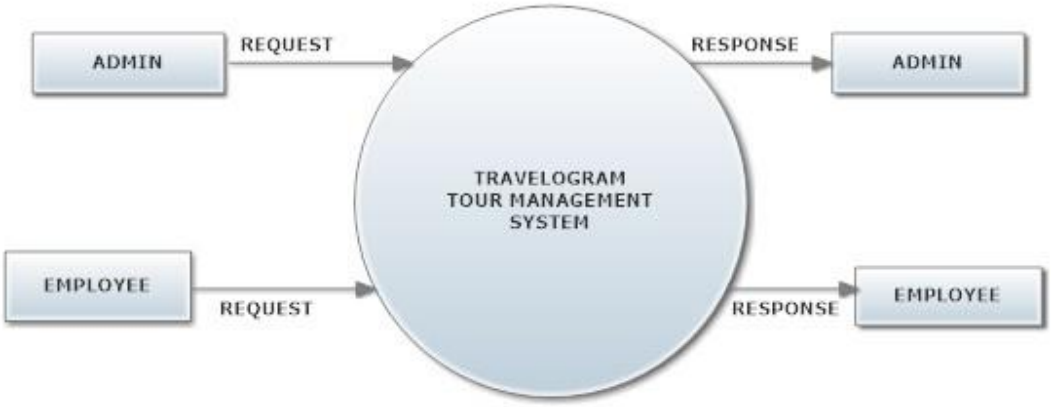


: **Circle**, this represents a process that transform

Incoming data and outgoing flow.

: **open Rectangle**, which shows data store.

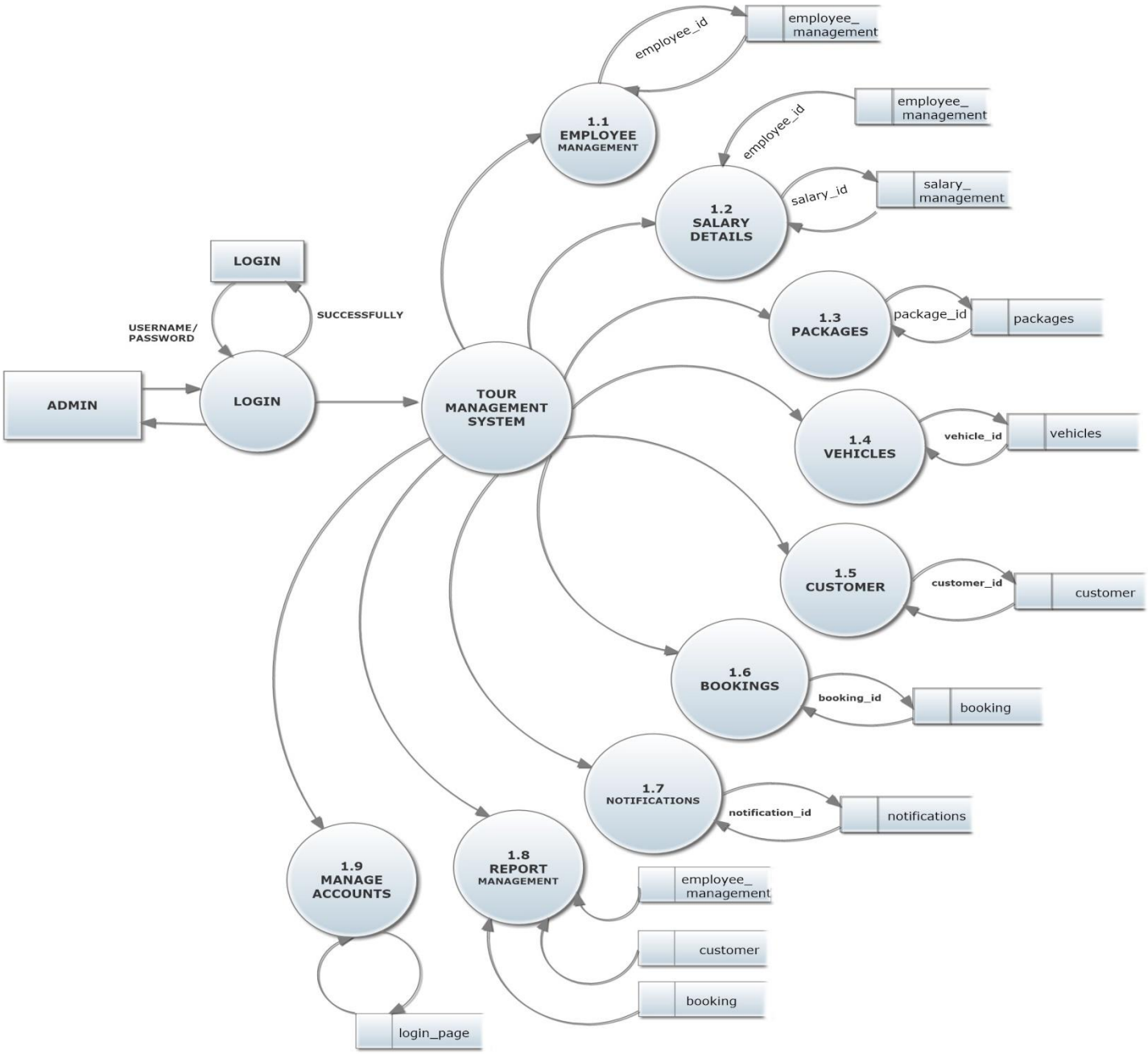
LEVEL 0



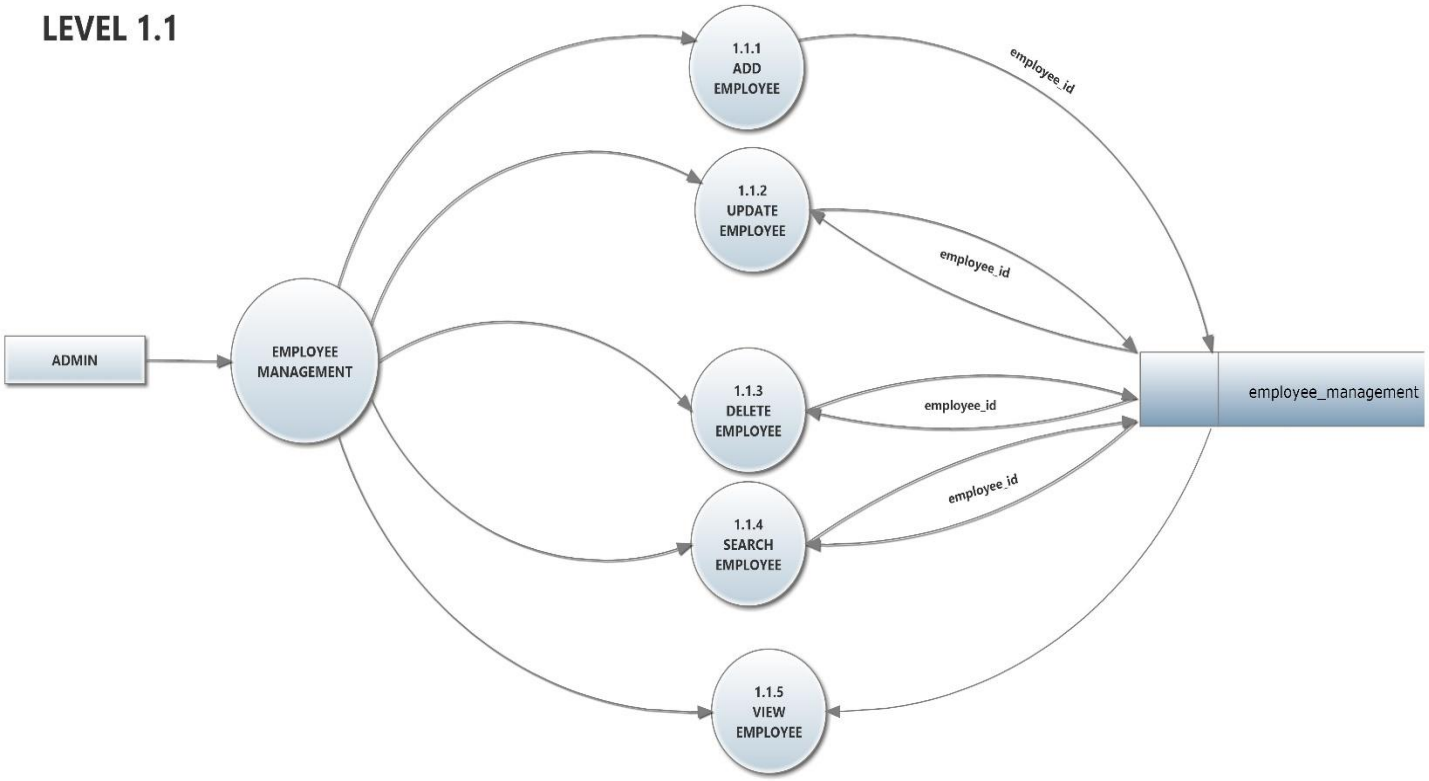
ADMINISTRATOR

Level 1

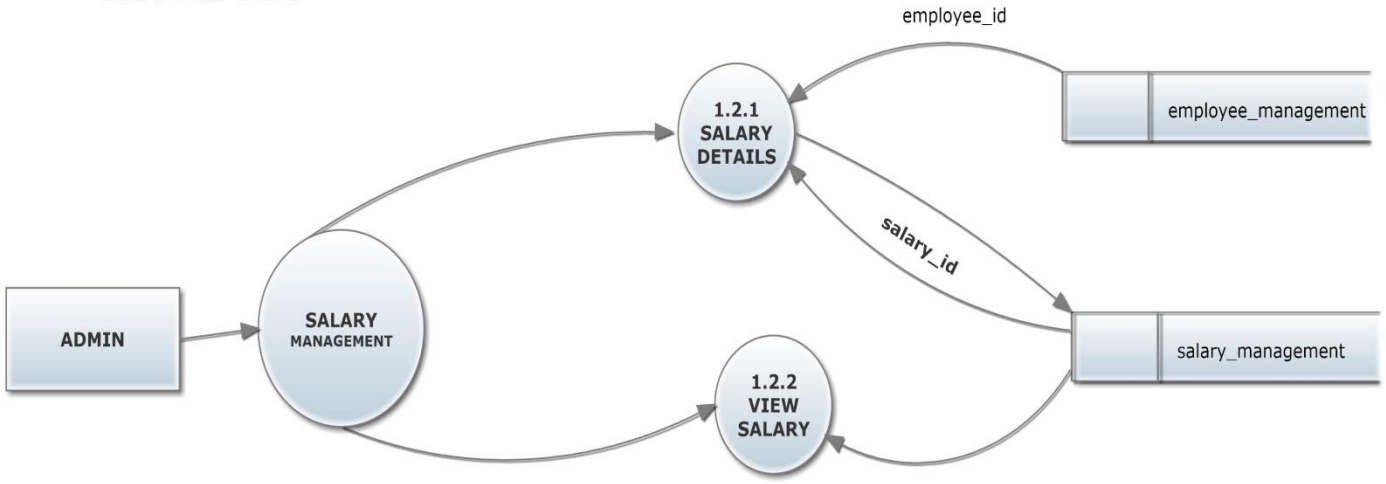
1



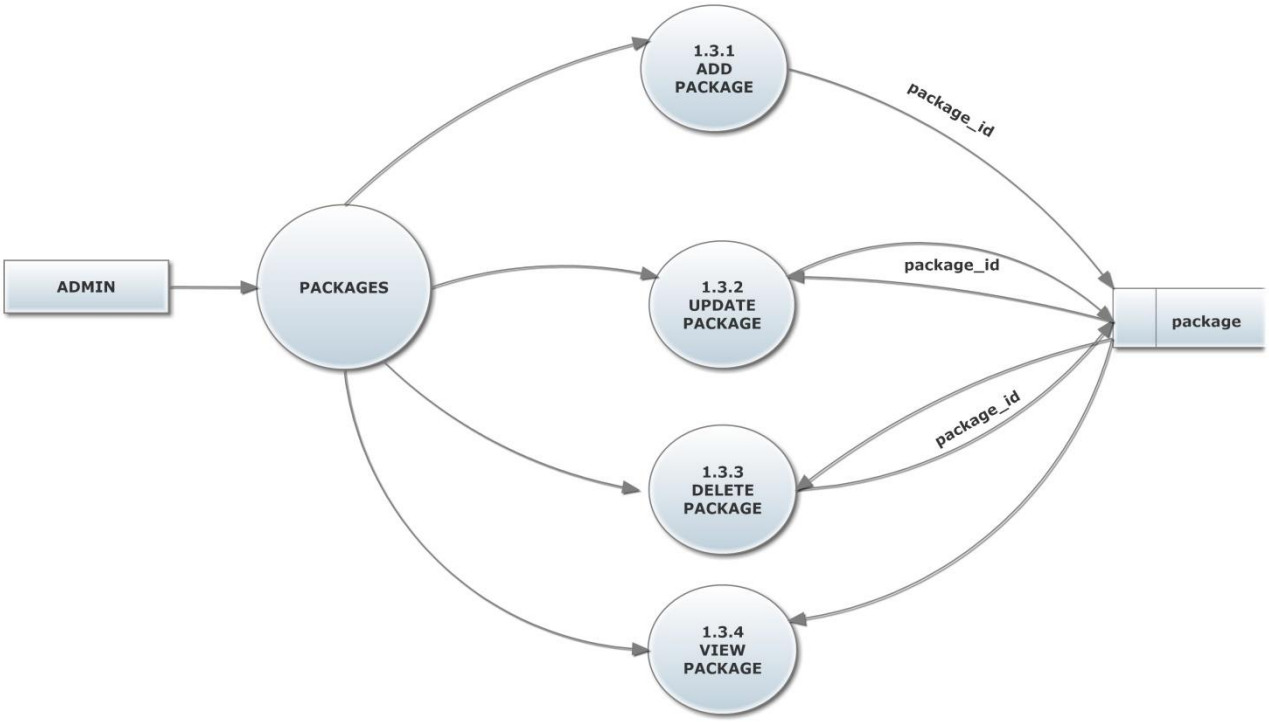
LEVEL 1.1



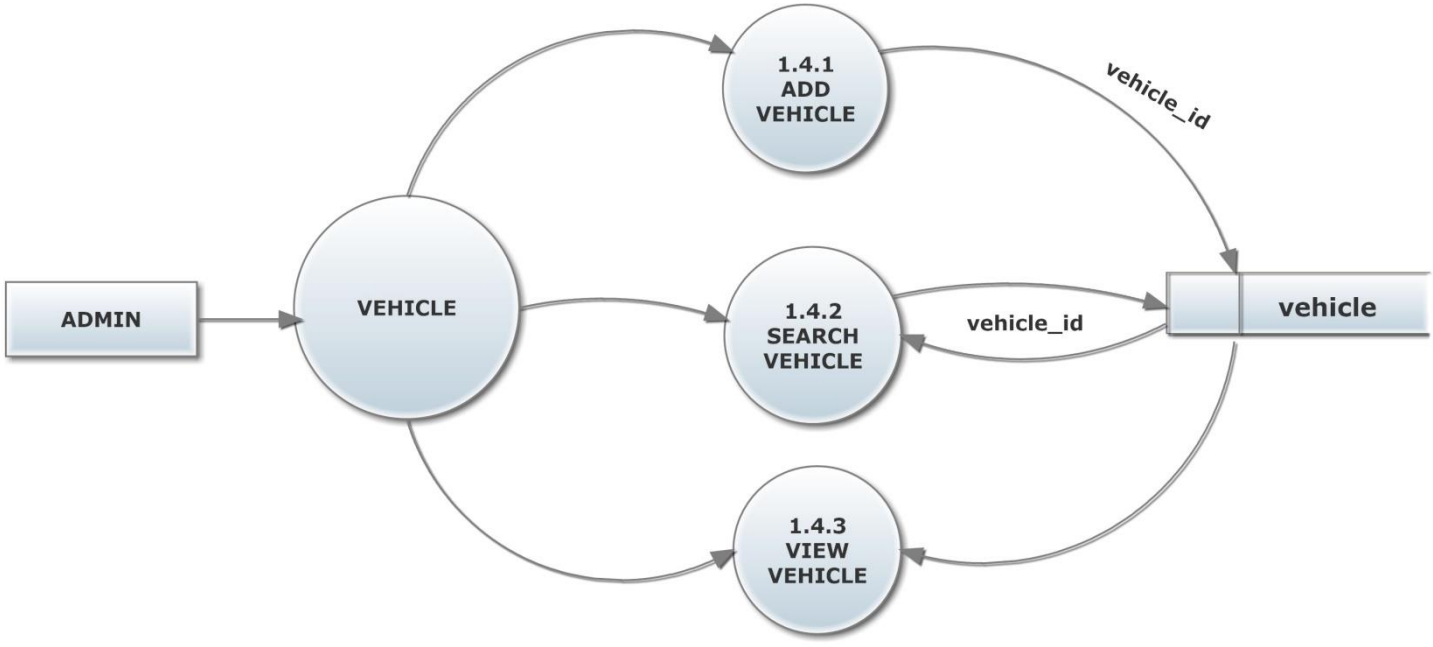
LEVEL 1.2



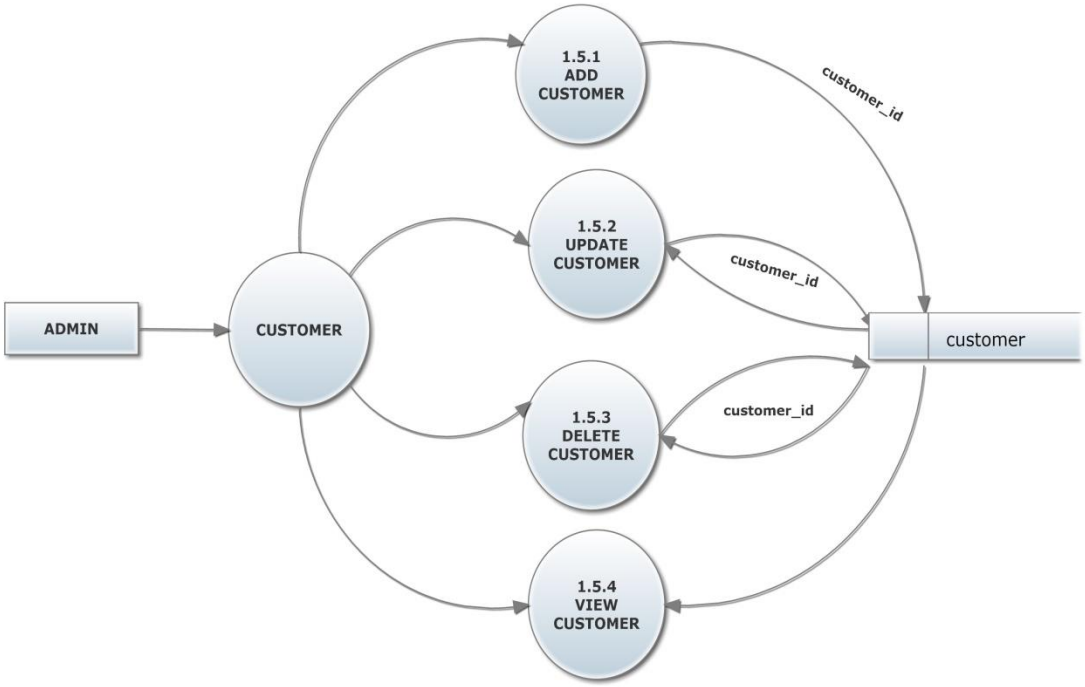
LEVEL 1.3



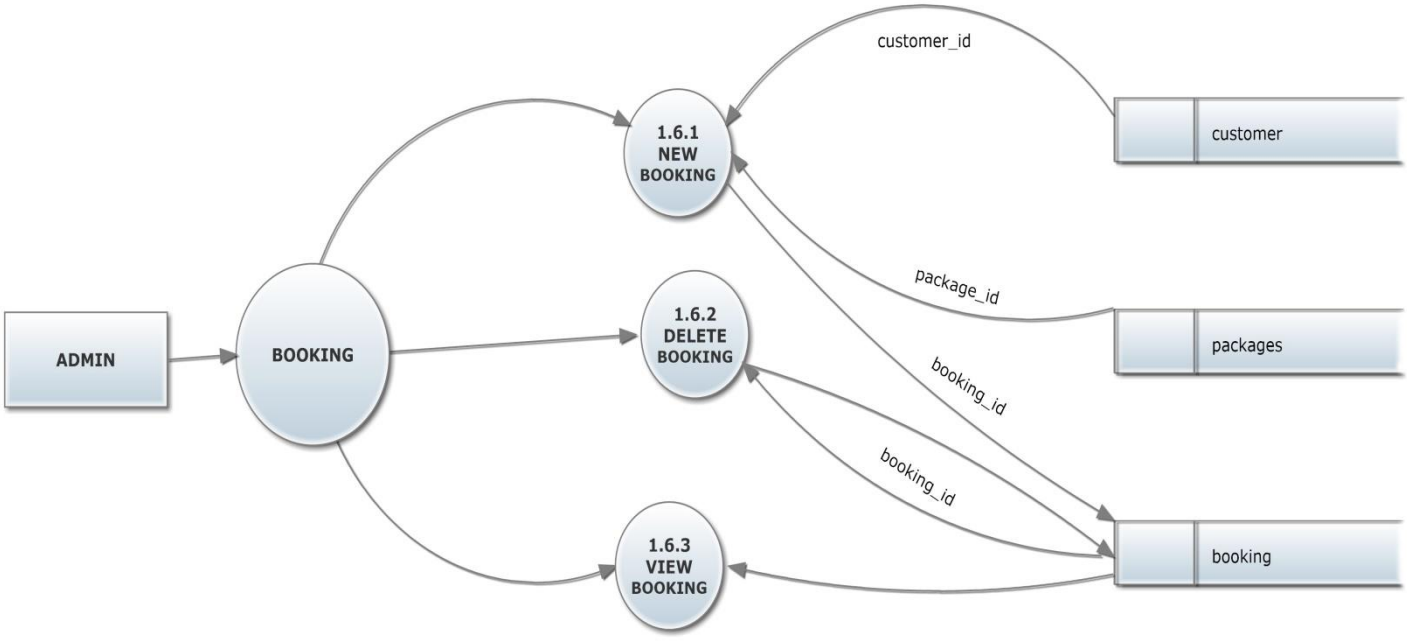
LEVEL 1.4



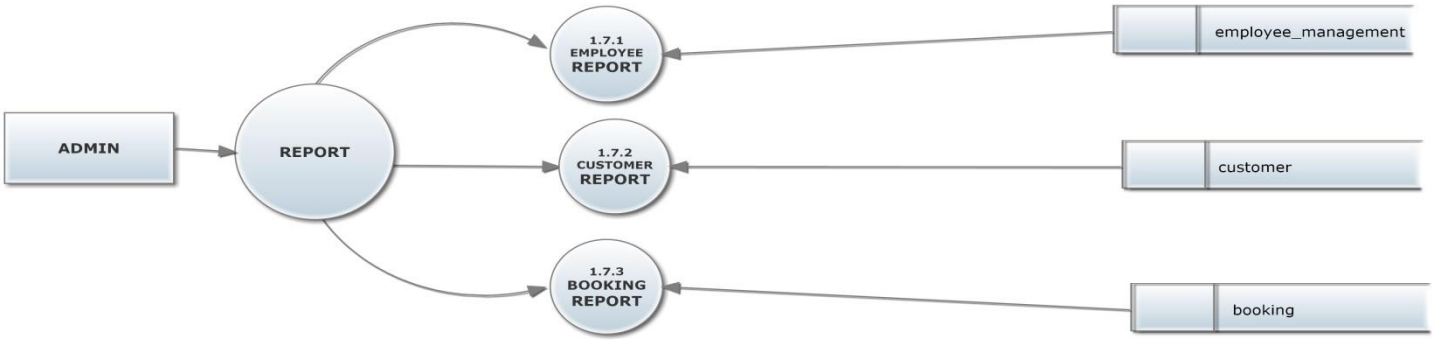
LEVEL 1.5



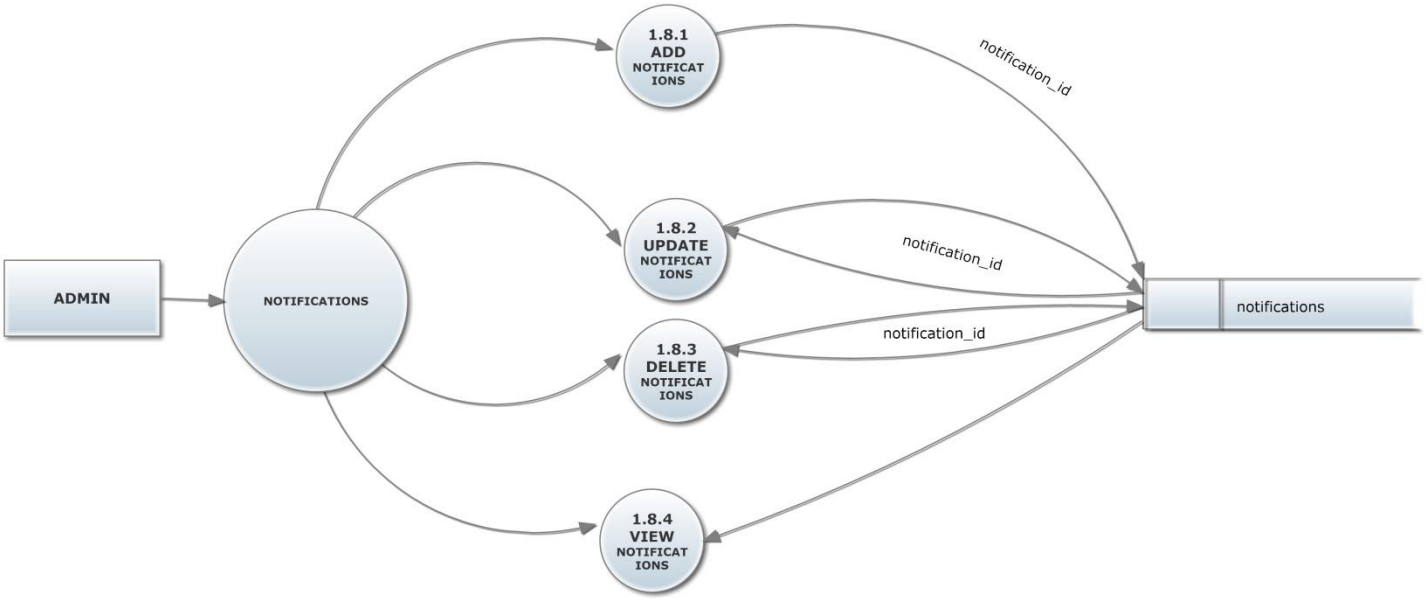
LEVEL 1.6



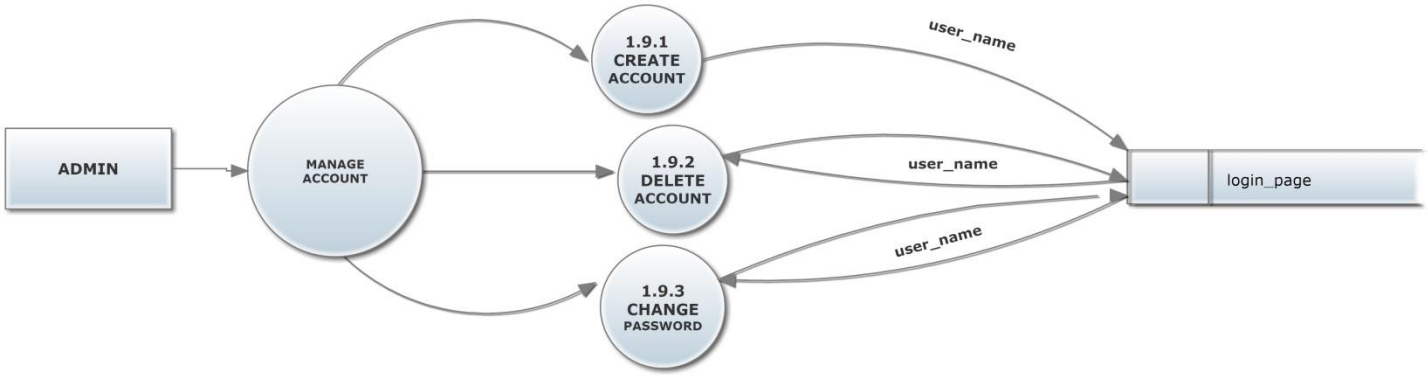
LEVEL 1.7



LEVEL 1.8

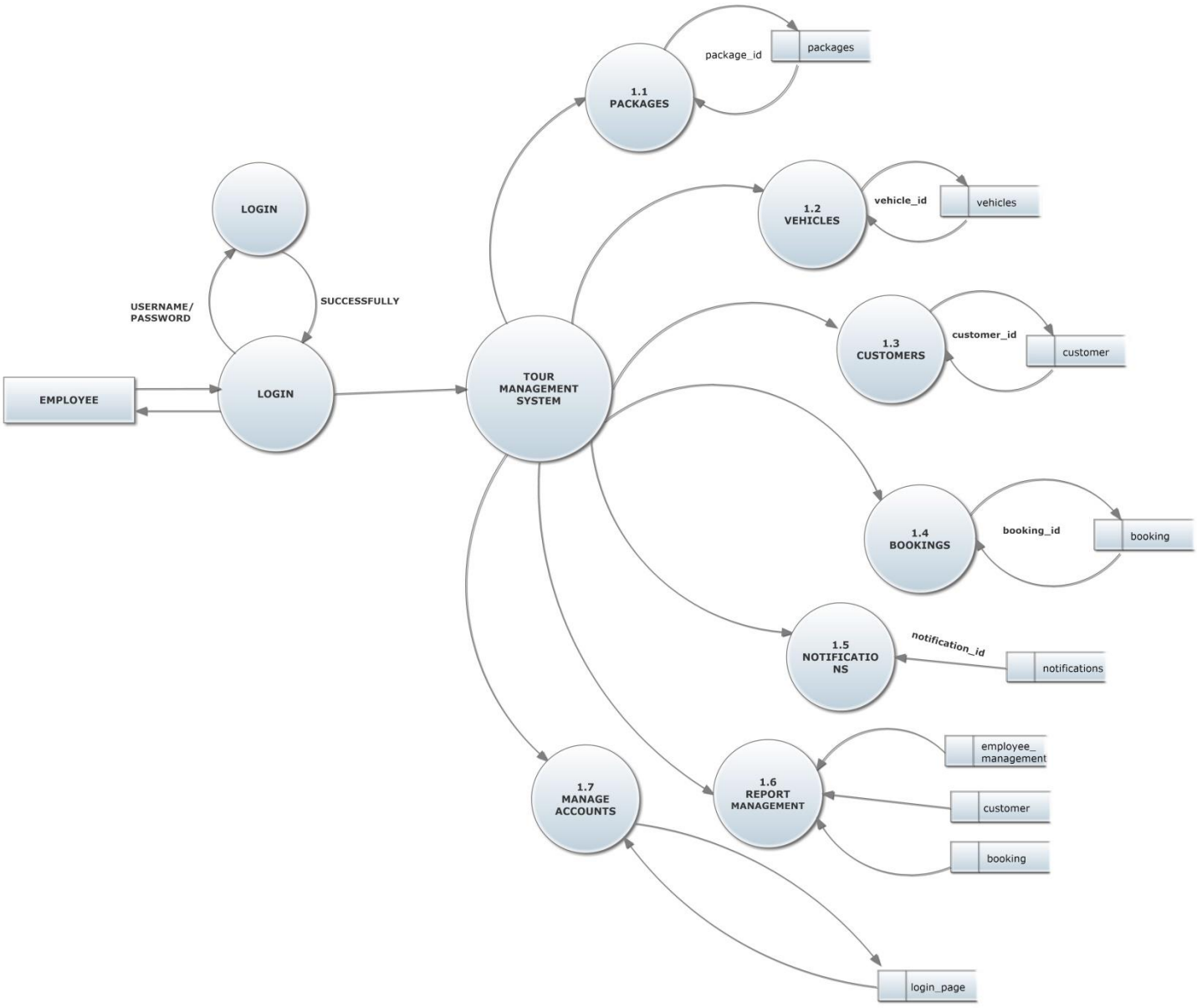


LEVEL 1.9

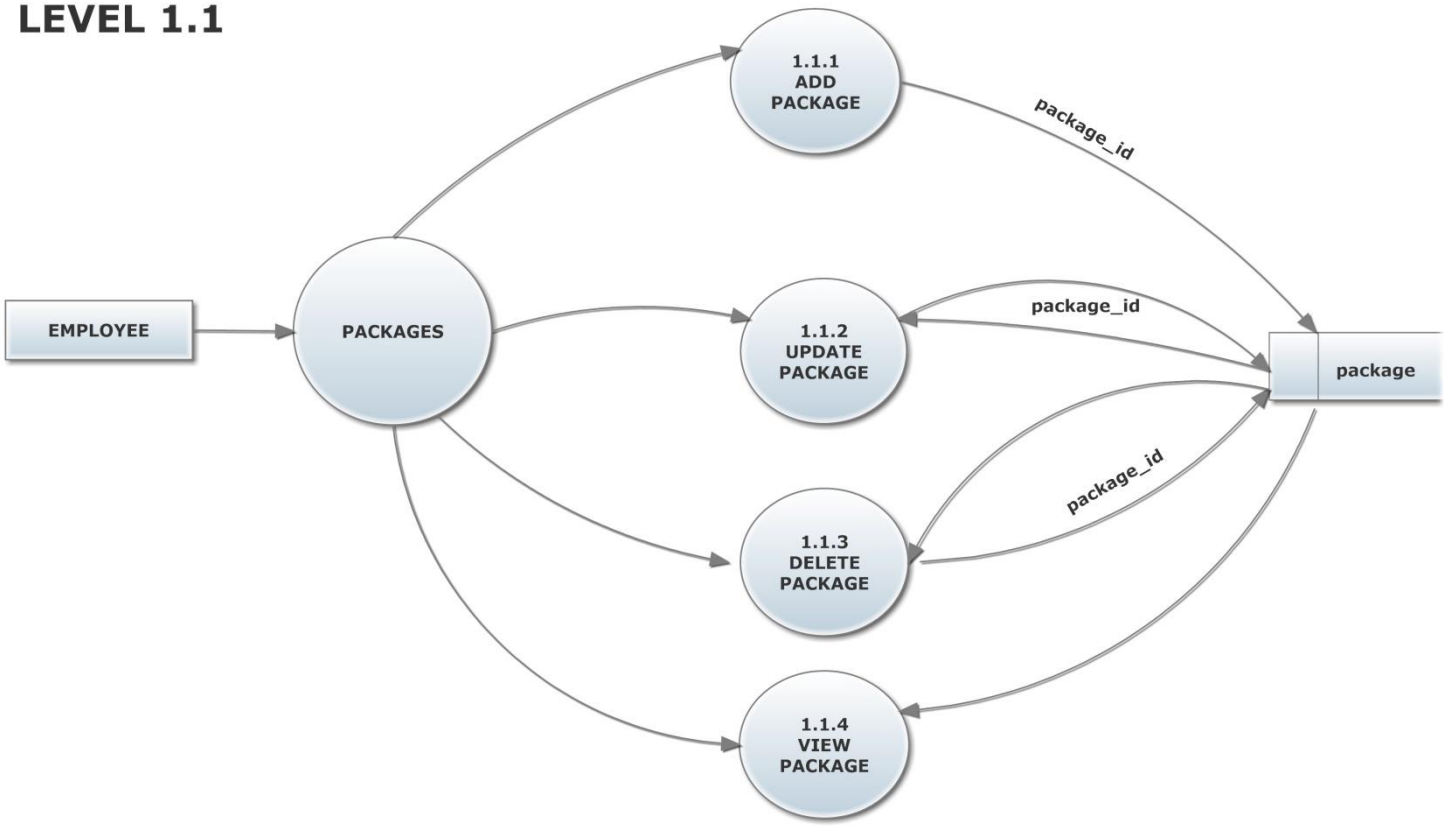


EMPLOYEE

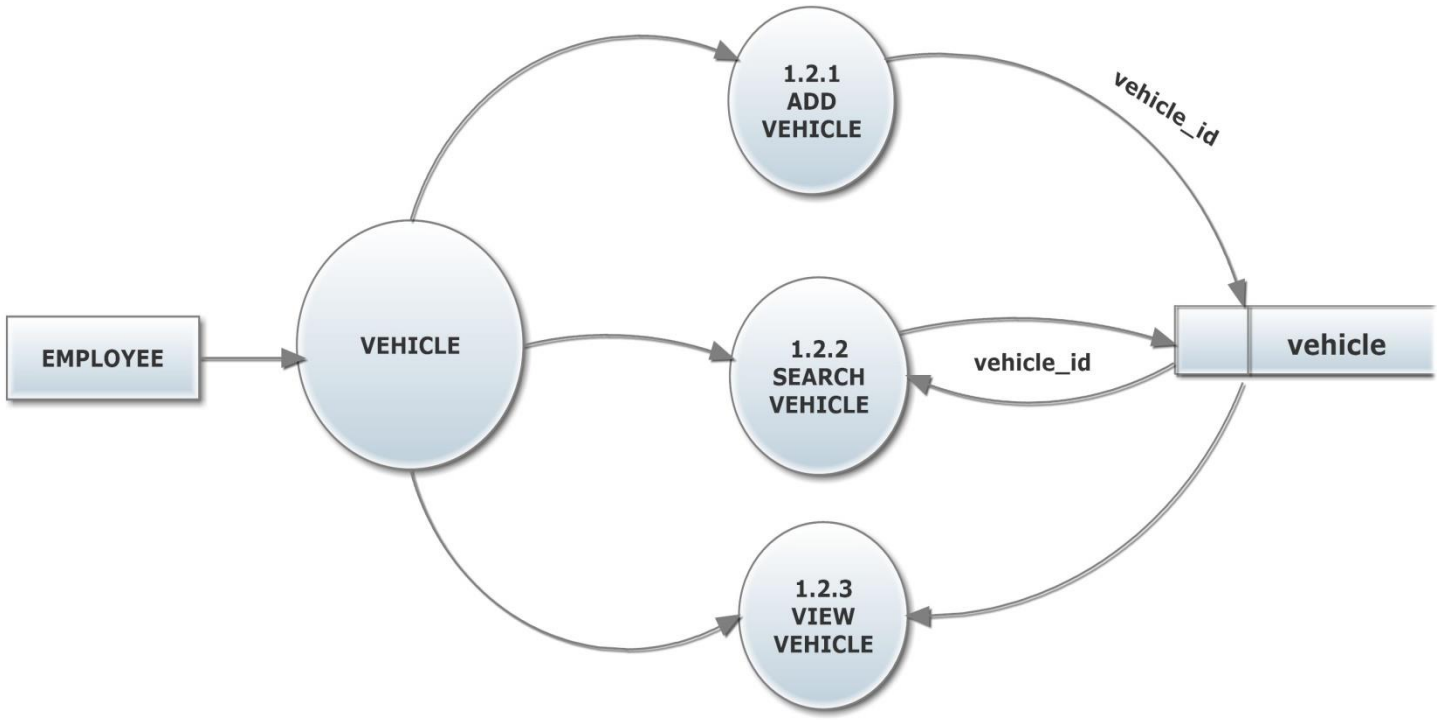
LEVEL 1



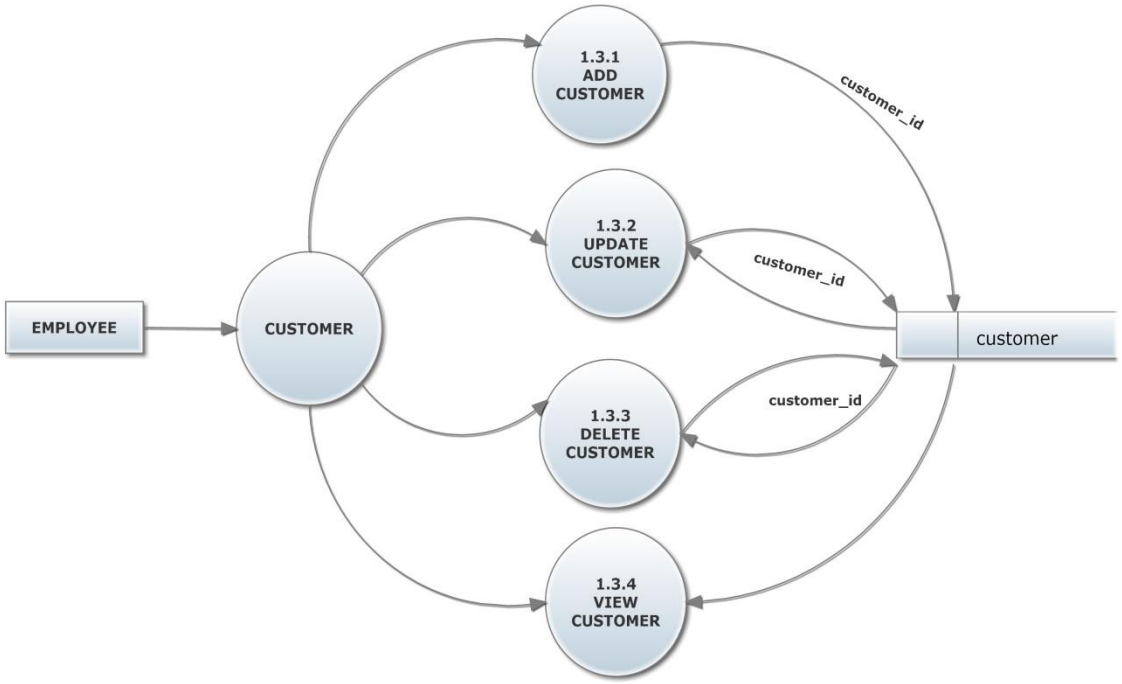
LEVEL 1.1



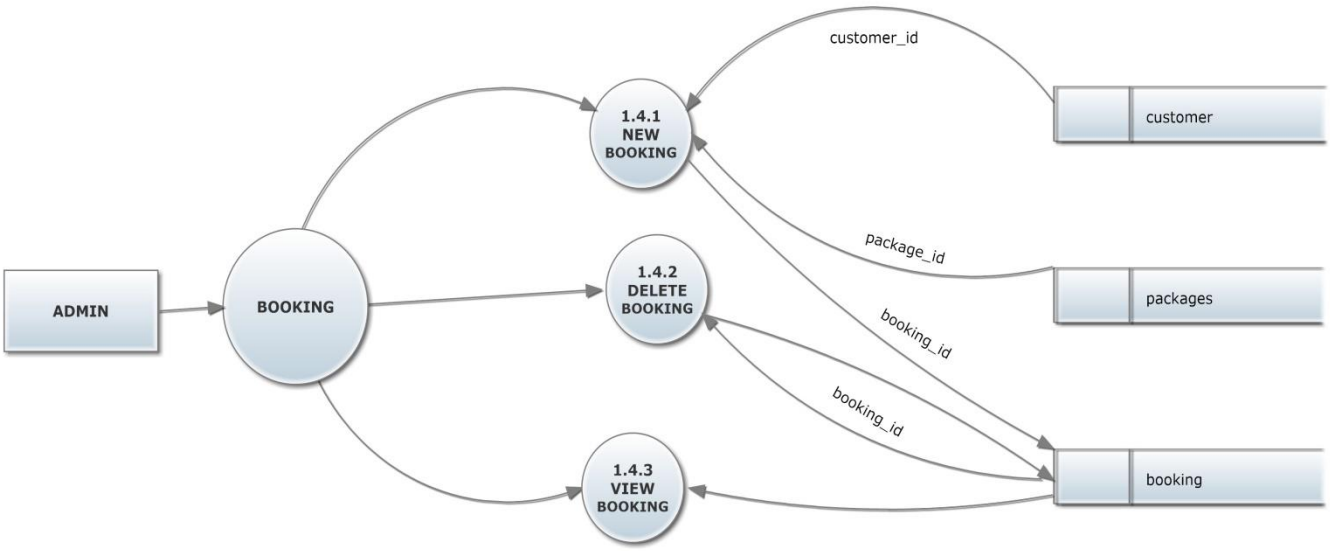
LEVEL 1.2



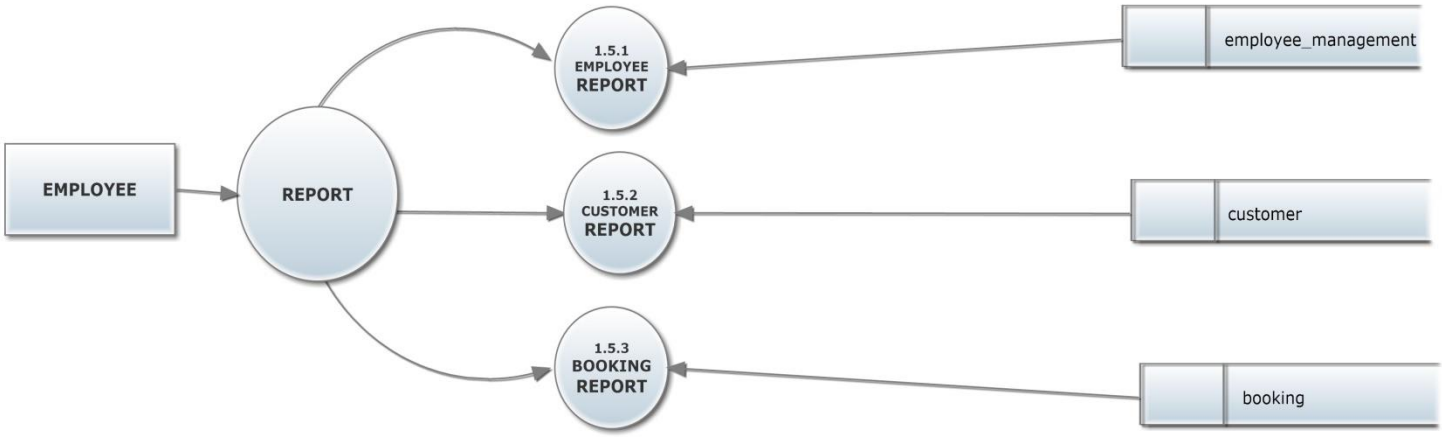
LEVEL 1.3



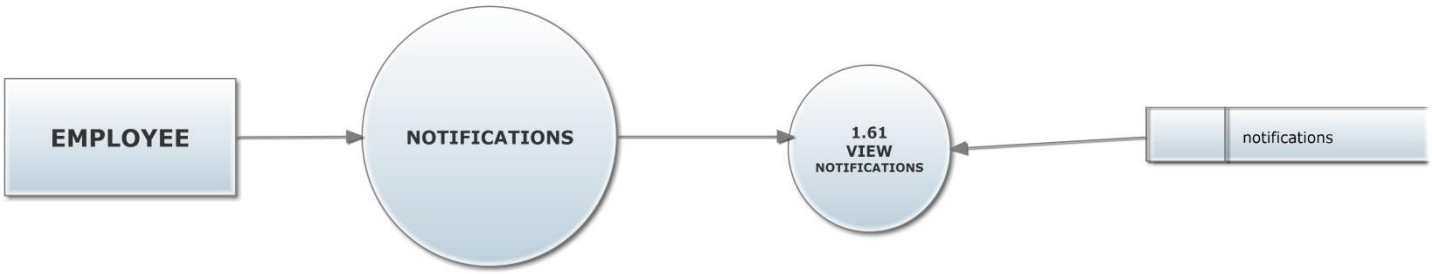
LEVEL 1.4



LEVEL 1.5

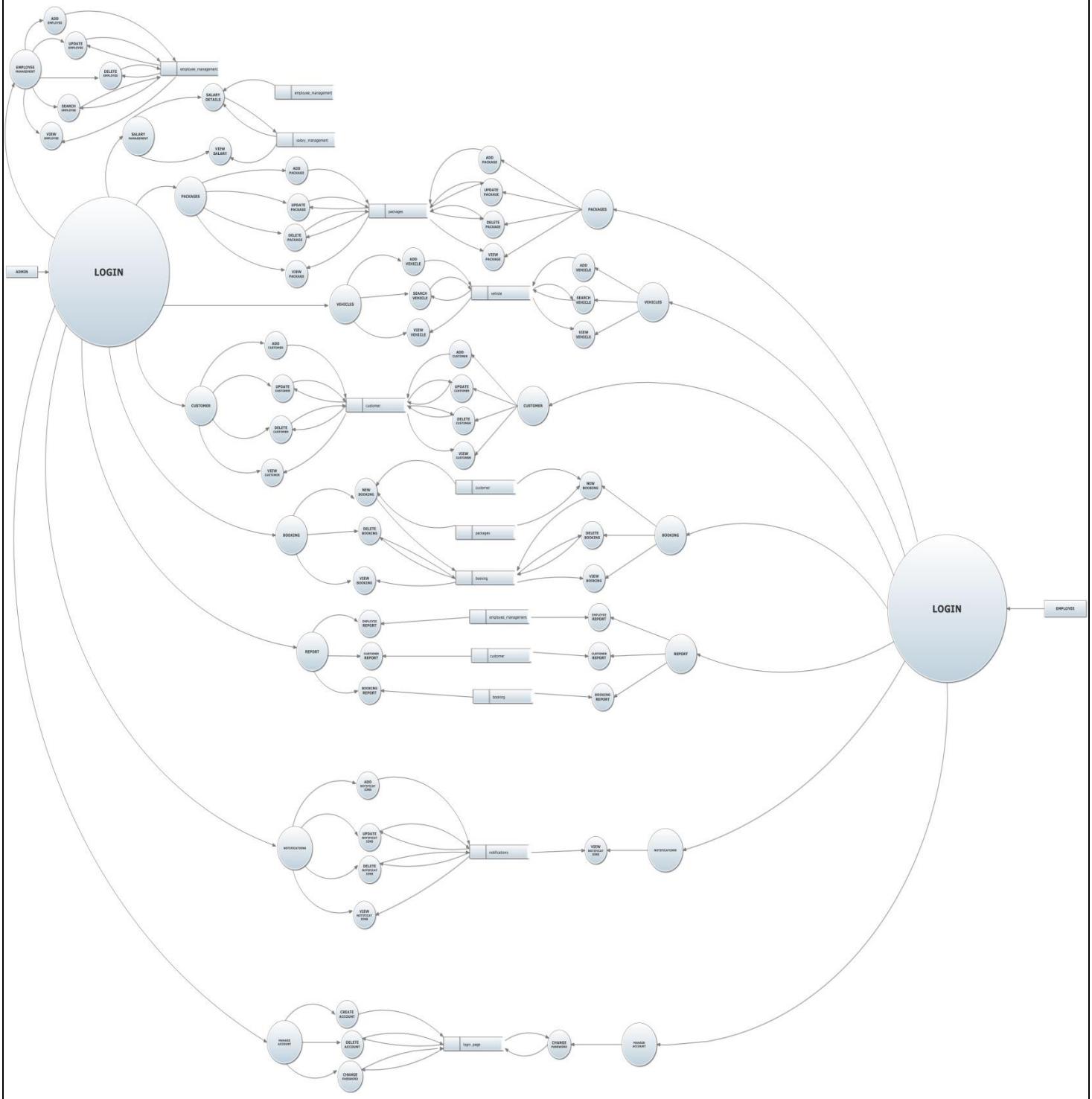


LEVEL 1.6



LEVEL 1.7

LEVEL 2


ADMIN
LOGIN
EMPLOYEE
EMPLOYEE MANAGEMENT
ADD EMPLOYEE
UPDATE EMPLOYEE
DELETE EMPLOYEE
SEARCH EMPLOYEE
VIEW EMPLOYEE
employee_management
SALARY DETAILS
SALARY MANAGEMENT
VIEW SALARY
salary_management
PACKAGES
ADD PACKAGE
UPDATE PACKAGE
DELETE PACKAGE
VIEW PACKAGE
packages
VEHICLES
ADD VEHICLE
SEARCH VEHICLE
VIEW VEHICLE
vehicle
CUSTOMER
ADD CUSTOMER
UPDATE CUSTOMER
DELETE CUSTOMER
VIEW CUSTOMER
customer
BOOKING
NEW BOOKING
DELETE BOOKING
VIEW BOOKING
booking
REPORT
EMPLOYEE REPORT
CUSTOMER REPORT
BOOKING REPORT
NOTIFICATIONS
ADD NOTIFICATIONS
UPDATE NOTIFICATIONS
DELETE NOTIFICATIONS
VIEW NOTIFICATIONS
notifications
MANAGE ACCOUNT
CREATE ACCOUNT
DELETE ACCOUNT
CHANGE PASSWORD
login_page
LOGIN
EMPLOYEE

3.3DATA BASE DESIGN

Table_1: LOGIN PAGE

Field Name	Data Type	Description	Constraints
user_name	varchar(20)	User Name	primary key
password	varchar(20)	Password	not null
account_type	varchar(30)	Account type	not null

Table_2: EMPLOYEE MANAGEMENT

Field Name	Data Type	Description	Constraints
employee_id	int	employee id	primary key
employee_ name	varchar(50)	employee name	not null
date_of_birth	varchar(25)	date of birth	not null
gender	varchar(10)	gender	not null
date_of _join	varchar(10)	date of join	not null
mobile	numeric(18,0)	Mobile number	not null
email_id	varchar(50)	email	not null
basic_pay	numeric(18,0)	basic pay	not null
house_ name	varchar(50)	house name	not null
village	varchar(50)	village	not null

city	varchar(40)	city	not null
town	varchar(50)	town	not null
pin_code	int	pin code	not null
states	varchar(40)	state	not null
country	varchar(40)	country	not null
email_id	varchar(50)	email	not null
pic	varchar(200)	picture	not null

Table_3: SALARY MANAGEMENT

Field Name	Data Type	Description	Constraints
salary_id	int	salary id	primary key
employee_id	int	employee_id	foreign key
employee_name	varchar(50)	employee_name	not null
basic_pay	varchar(50)	basic pay	not null
da	varchar(50)	da	not null
hra	varchar(10)	hra	not null
providend_ fund	varchar(10)	providend fund	not null
total_salary	varchar(10)	total salary	not null

Table_4: PACKAGES

Field Name	Data Type	Description	Constraints
package_id	int	package id	primary key
state	varchar(20)	state	not null

spot	varchar(30)	spot	not null
distance	float	distance	not null
car_charges	float	Car charges	not null
bus_charges	float	Bus charges	not null
stay_cost	float	Stay cost	not null
hotel1	varchar(30)	Hotel 1	not null
hotel2	varchar(30)	Hotel 2	not null
hotel3	varchar(30)	Hotel 3	not null

Table_5: VEHICLE

Field Name	Data Type	Description	Constraints
vehicle_id	int	vehicle id	primary key
vehicle_type	varchar(20)	Vehicle type	not null
fuel_type	varchar(50)	Fuel type	not null
company_name	varchar(20)	Company name	not null
vehicle_model	varchar(15)	Vehicle model	not null
registration_date	varchar(15)	Registration date	not null
vehicle_number	varchar(50)	Vehicle number	not null
total_seats	int	Total seats	not null
fuel_capacity	varchar(15)	Fuel capacity	not null

Table_6: NOTIFICATIONS

Field Name	Data Type	Description	Constraints
notification_id	int	notification id	primary key
purpose	varchar(100)	purpose	not null
notification_date	date	notification date	not null
notification_time_H	int	notification time	not null
notification_time_M	int	notification time	not null
am_pm	varchar(15)	am and pm	not null
venue	varchar(20)	venue	not null

Table_7: CUSTOMER

Field Name	Data Type	Description	Constraints
customer_id	int	customer id	primary key
customer_name	varchar(50)	first name	not null
gender	varchar(10)	gender	not null
street	varchar(50)	street	not null
city	varchar(40)	City	not null
state	varchar(40)	state	not null
pin_code	int	pin code	not null
mobile	numeric(18,0)	mobile	not null
email_id	varchar(50)	email id	not null

Table_8: BOOKING

Field Name	Data Type	Description	Constraints
booking_id	int	booking id	primary key
customer_id	int	customer id	foreign key
package_id	int	Package id	foreign key
customer_name	varchar(50)	Customer name	not null
mobile	int	mobile	not null
State	varchar(50)	state	not null
hotel	varchar(50)	hotel	not null
number_of persons	int	Number of persons	not null
boarding_date	varchar(10)	Boarding date	not null
email_id	varchar(20)	email id	not null
spot	varchar(30)	Spot	not null

SYSTEM DEVELOPMENT

4.1 CODING

MDI FORM

```
Private Sub ADDCUSTOMER_Click()

add_customer.Show

End Sub


Private Sub ADDEMPLOYEE_Click()

add_employee.Show

End Sub


Private Sub ADDNOTIFICATION_Click()

add_notification.Show

End Sub


Private Sub ADDPACKAGE_Click()

add_package.Show

End Sub


Private Sub ADDVEHICLES_Click()

vehicle_registration.Show

End Sub
```

```
Private Sub BOOKINGREPORT_Click()
booking_report.Show
End Sub

Private Sub CHANGEPASSWORD_Click()
change_password.Show
End Sub

Private Sub CREATEACCOUNT_Click()
create_account.Show
End Sub

Private Sub CUSTOMERREPORT_Click()
customer_report.Show
End Sub

Private Sub DELETEACCOUNT_Click()
delete_account.Show
End Sub

Private Sub DELETEBOOKING_Click()
delete_booking.Show
End Sub
```

```
Private Sub DELETECUSTOMER_Click()
delete_customer.Show
End Sub

Private Sub DELETEEMPLOYEE_Click()
delete_employee.Show
End Sub

Private Sub DELETENOTIFICATION_Click()
delete_nitification.Show
End Sub

Private Sub DELETEPACKAGES_Click()
DELETE_PACKAGE.Show
End Sub

Private Sub EMPLOYEEREPORT_Click()
employee_report.Show
End Sub

Private Sub LOGOUT_Click()
Unload Me
```

```
Login.Show
End Sub

Private Sub NEWBOOKING_Click()
new_booking.Show
End Sub

Private Sub SALARYDETAILS_Click()
salary_details.Show
End Sub

Private Sub SEARCHEMPLOYEE_Click()
search_employee.Show
End Sub

Private Sub SEARCHVEHICLES_Click()
search_vehicle.Show
End Sub

Private Sub UPDATECUSTOMER_Click()
update_customer.Show
End Sub
```

```
Private Sub UPDATEEMPLOYEE_Click()
update_employee.Show
End Sub

Private Sub UPDATENOTIFICATION_Click()
uodate_notification.Show
End Sub

Private Sub UPDATEPACKAGE_Click()
update_package.Show
End Sub

Private Sub VIEWBOOKINGS_Click()
booking_view.Show
End Sub

Private Sub VIEWCUSTOMER_Click()
customer_view.Show
End Sub

Private Sub VIEWEMPLOYEE_Click()
employee_view.Show
End Sub
```

```
Private Sub VIEWNOTIFICATION_Click()
notificatiom_view.Show
End Sub

Private Sub VIEWPACKAGE_Click()
package_view.Show
End Sub

Private Sub VIEWSALARY_Click()
salary_view.Show
End Sub

Private Sub VIEWVEHICLES_Click()
vehicle_view.Show
End Sub
```

LOGIN PAGE

```
Option Explicit
Public con As New ADODB.Connection
Public rs As New ADODB.Recordset
Public cmd As New ADODB.Command
```

```
Public str As String

Public str1 As String

Dim status As String

Dim t As String

Public SQL As String

Public Sub connect()

con.Provider = "SQLOLEDB"

str1 = "server=DESKTOP-HOTR91D\SQLEXPRESS;database=master;trusted_connection=yes"

con.Open str1

End Sub


Private Sub command1_Click()

Call connect

str1 = "select * from login_page"

rs.Open str1, con, adOpenKeyset

status = False

Do

If rs.EOF Then

MsgBox "invalid login , user name and password are not correct"

con.Close

End If

If rs("user_name").Value = Text1.Text And rs("password").Value = Text2.Text Then

status = True
```

```
t = rs("account_type").Value
Exit Do
Else
rs.MoveNext
End If
Loop Until rs.EOF
If status = True And t = "ADMIN" Then
MDIForm1.Show
Else
If status = True And t = "EMPLOYEE" Then
MDIForm1.Show
MDIForm1.ADDEMPLOYEE.Visible = False
MDIForm1.DELETEEMPLOYEE.Visible = False
MDIForm1.UPDATEEMPLOYEE.Visible = False
MDIForm1.SEARCHEMPLOYEE.Visible = False
MDIForm1.EMPLOYEEMANAGEMENT.Visible = False
MDIForm1.SALARYMANAGEMENT.Visible = False
MDIForm1.ADDNOTIFICATION.Visible = False
MDIForm1.CREATEACCOUNT.Visible = False
MDIForm1.DELETEACCOUNT.Visible = False
MDIForm1.DELETENOTIFICATION.Visible = False
MDIForm1.UPDATENOTIFICATION.Visible = False
Else
```

```
MsgBox "invalid username or password"
End If
End If
con.Close
End Sub
Private Sub Command2_Click()
Text1.Text = ""
Text2.Text = ""
End Sub
```

ADD EMPLOYEE

```
Option Explicit
Public con As New ADODB.Connection
Public rs As New ADODB.Recordset
Public cmd As New ADODB.Command
Public str As String
Public str1 As String
Public str2 As String

Public Sub connect()
con.Provider = "sqloledb"
str1 = "server=DESKTOP-HOTR91D\SQLEXPRESS;database=master;trusted_connection=yes"
```

```
con.Open str1
End Sub

Private Sub Image1_Click()
CommonDialog1.ShowOpen
str = CommonDialog1.FileName
Picture1.Picture = LoadPicture(str)
End Sub

Private Sub Form_Load()
Dim n As Integer
Call connect
str1 = "select * from employee_management"
rs.Open str1, con, adOpenKeyset
con.Execute str1
If rs.RecordCount = 0 Then
Text1.Text = "1"
Else
rs.MoveLast
n = rs("employee_id").Value
Text1.Text = n + 1
End If
rs.Close
```

```
con.Close

End Sub


Private Sub Image3_Click()

Unload Me

End Sub


Private Sub Image2_Click()

If Text2.Text = "" Then

MsgBox "Please Enter Employee Name !", vbExclamation, "Add Employee"

Exit Sub

End If


If Option1.Value = False And Option2.Value = False Then

MsgBox "Please Select Gender !", vbExclamation, "Add Employee"

Exit Sub

End If

If Text3.Text = "" Then

MsgBox "Please Enter Mobile Number !", vbExclamation, "Add Employee"

Exit Sub

End If


If Text4.Text = "" Then
```

```
MsgBox "Please Enter E-Mail ID !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text5.Text = "" Then
MsgBox "Please Enter Basic Pay !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text6.Text = "" Then
MsgBox "Please Enter House Name !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text7.Text = "" Then
MsgBox "Please Enter Village !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text8.Text = "" Then
MsgBox "Please Enter City !", vbExclamation, "Add Employee"
Exit Sub
End If
```

```
If Text9.Text = "" Then
MsgBox "Please Enter Town !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text10.Text = "" Then
MsgBox "Please Enter PIN Code !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text11.Text = "" Then
MsgBox "Please Enter State !", vbExclamation, "Add Employee"
Exit Sub
End If

If Text12.Text = "" Then
MsgBox "Please Enter Country !", vbExclamation, "Add Employee"
Exit Sub
End If

If Picture1.Picture = 0 Then
MsgBox "Please Upload Employees Photo !", vbExclamation, "Add Employee"
```

```
Exit Sub

End If

If Option1.Value = True Then


Call connect

str1 = "insert into
employee_management(employee_id,employee_name,date_of_birth,gender,date_of_join,
mobile,email_id,basic_pay,house_name,village,city,town,pin_code,states,country,pic)values
(" & Text1.Text & ",'" & Text2.Text & "','" & DTPicker1 & "','" & Option1.Caption & "','" &
DTPicker2 & "'," & Text3.Text & ",'" & Text4.Text & "'," & Text5.Text & ",'" & Text6.Text &
"','" & Text7.Text & "','" & Text8.Text & "','" & Text9.Text & "'," & Text10.Text & ",'" &
Text11.Text & "','" & Text12.Text & "','" & str & "')"


con.Execute str1

MsgBox "Details saved successfully...", vbInformation, "Save Employee Details"


Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Text8.Text = ""

Text9.Text = ""

Text10.Text = ""
```

```
Text11.Text = ""

Text12.Text = ""

Picture1.Picture = Nothing

Option1.Value = False

'Option2.Value = False

con.Close


End If

If Option2.Value = True Then

Call connect

str1 = "insert into
employee_management(employee_id,employee_name,date_of_birth,gender,date_of_join,
mobile,email_id,basic_pay,house_name,village,city,town,pin_code,states,country,pic)values
(" & Text1.Text & ",'" & Text2.Text & "','" & DTPicker1 & "','" & Option2.Caption & "','" &
DTPicker2 & "'," & Text3.Text & ",'" & Text4.Text & "'," & Text5.Text & ",'" & Text6.Text &
"','" & Text7.Text & "','" & Text8.Text & "','" & Text9.Text & "'," & Text10.Text & ",'" &
Text11.Text & "','" & Text12.Text & "','" & str & "')

con.Execute str1

MsgBox "Details saved successfully...", vbInformation, "Save EmployeeDetails"

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""
```

```
Text8.Text = ""
Text9.Text = ""
Text10.Text = ""
Text11.Text = ""
Text12.Text = ""
Option2.Value = False
Picture1.Picture = Nothing
con.Close
End If
''id
Call connect
str1 = "select * from employee_management"
rs.Open str1, con, adOpenKeyset
con.Execute str1
Dim n As Integer
If rs.RecordCount = 0 Then
Text1.Text = "1"
Else
rs.MoveLast
n = rs("employee_id").Value
Text1.Text = n + 1
End If
rs.Close
```

```
con.Close

End Sub

Private Sub Text5_Validate(cancel As Boolean)

If Not (IsNumeric(Text5.Text)) Then

MsgBox " Please Enter Basic Pay Numerically !", vbExclamation, "Add Employee"

End If

End Sub


Private Sub Text10_Validate(cancel As Boolean)

'pin

If Not (IsNumeric(Text10.Text)) Then

MsgBox " Please Enter PIN Code Numerically !", vbExclamation, "Add Employee"

End If

If Len(Text10.Text) < 6 Or Len(Text10.Text) > 6 Then

MsgBox "Please Enter a Valid PIN Code !", vbExclamation, "Add Employee"

End If

End Sub


Private Sub Text3_Validate(cancel As Boolean)

'mobile

If Not (IsNumeric(Text3.Text)) Then

MsgBox " Please Enter Phone Number Numerically !", vbExclamation, "Add Employee"

End If
```

```
If Len(Text3.Text) < 10 Or Len(Text3.Text) > 10 Then

MsgBox "Enter the phone number in 10 digits !", vbExclamation, "Add Employee"

End If

End Sub
```

SALARY DETAILS

```
Private Sub Combo1_click()

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Call connect

str1 = "select employee_name,basic_pay from employee_management where
employee_id=" & Combo1.Text & ""

rs.Open str1, con, adOpenKeyset

Do While Not rs.EOF

Text2.Text = rs.Fields(0)

Text3.Text = rs.Fields(1)

rs.MoveNext

Loop

rs.Close
```

```
str2 = "select da,hra,providend_fund,total_salary from salary_management where
employee_id=" & Combo1.Text & ""
rs.Open str2, con, adOpenKeyset
Do While Not rs.EOF

Text4.Text = rs.Fields("da").Value
Text5.Text = rs.Fields("hra").Value
Text6.Text = rs.Fields("providend_fund").Value
Text7.Text = rs.Fields("total_salary").Value
rs.MoveNext
Loop
rs.Close
con.Close
End Sub

Private Sub Form_Load()
If con.State = adStateOpen Then
rs.Close
con.Close
End If

Call connect
str1 = "select * from salary_management"
rs.Open str1, con, adOpenKeyset
```

```
con.Execute str1
If rs.RecordCount = 0 Then
Text1.Text = "1"
Else
rs.MoveLast
n = rs("salary_id").Value
Text1.Text = n + 1
End If
rs.Close
con.Close

Call connect
str1 = "select employee_id from employee_management"
rs.Open str1, con, adOpenKeyset
con.Execute str1
Do While Not rs.EOF
Combo1.AddItem (rs.Fields(0))
rs.MoveNext
Loop
rs.Close
con.Close
End Sub
```

```
Private Sub Image3_Click()

Unload Me

End Sub


Private Sub total_salary_Click()

Text7.Text = Val(Text3.Text) + Val(Text4.Text) + Val(Text5.Text) + Val(Text6.Text)

End Sub


Private Sub Image_save_Click()

If Combo1.Text = "" Then

MsgBox "Please choose Employee ID", vbExclamation, "Salary Management"

Exit Sub

End If


If Text4.Text = "" Then

MsgBox "Please Enter Da !", vbExclamation, "Salary Management"

Exit Sub

End If


If Text5.Text = "" Then

MsgBox "Please Enter  HRA !", vbExclamation, "Salary Management"

Exit Sub

End If
```

```
If Text6.Text = "" Then
MsgBox "Please Enter PF !", vbExclamation, "Salary Managemet"
Exit Sub
End If

If Text7.Text = "" Then
MsgBox "Please calculate Total salary", vbExclamation, "Salary Management"
Exit Sub
End If

Call connect
str = "insert into
salary_management(salary_id,employee_id,employee_name,basic_pay,hra,da,providend_f
und,total_salary) values (" & Text1.Text & "," & Combo1.Text & ",'" & Text2.Text & "','" &
Text3.Text & "'," & Text4.Text & ",'" & Text5.Text & "'," & Text6.Text & "," & Text7.Text & ")"
con.Execute str
MsgBox "Salary saved successfully...", vbInformation, "Salary Management"

Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
```

```
Text7.Text = ""
Combo1.Text = ""
con.Close

Call connect
str1 = "select * from salary_management"
rs.Open str1, con, adOpenKeyset
con.Execute str1
If rs.RecordCount = 0 Then
Text1.Text = "1"
Else
rs.MoveLast
n = rs("salary_id").Value
Text1.Text = n + 1
End If
rs.Close
con.Close
End Sub

'Private Sub Command8_Click()
'Call connect
'str1 = " update salary set da= ' " & Text2.Text & " ',hra=' " & Text3.Text & " ',pf=' " &
Text4.Text & " ' where eid=' " & Combo1.Text & "'"
'con.Execute str1
```

```
'MsgBox " succesfully updated", vbInformation

'con.Close

'Adodc1.Recordset.Update

'End Sub


Private Sub Image5_Click()

Call connect

confirm = MsgBox("Are you sure you want to delete this Salary details ?", vbYesNo, "Delete this salary details")

If confirm = vbYes Then

str1 = "delete from salary_management where employee_id= '" & Combo1.Text & "' "

con.Execute str1

MsgBox "successfully deleted ", vbInformation

End If

con.Close

End Sub
```

UPDATE CUSTOMER

```
Private Sub cancel_Click()

Unload Me

End Sub

Private Sub Combo1_click()
```

```
Call connect
str1 = "select * from customer where customer_id=" & Combo1.Text & ""
rs.Open str1, con, adOpenKeyset
Do While Not rs.EOF
If rs.Fields("gender").Value = Option1.Caption Then
Option1.Value = True
Else
Option2.Value = True
End If
Text1.Text = rs.Fields("customer_name").Value
Text2.Text = rs.Fields("street").Value
Text3.Text = rs.Fields("city").Value
Text4.Text = rs.Fields("states").Value
Text5.Text = rs.Fields("pin_code").Value
Text6.Text = rs.Fields("email_id").Value
Text7.Text = rs.Fields("mobile").Value
rs.MoveNext
Loop
rs.Close
con.Close
End Sub
Private Sub Form_Load()
If con.State = adStateOpen Then
```

```
rs.Close

con.Close

End If

Call connect

str1 = "select customer_id from customer"

rs.Open str1, con, adOpenKeyset

con.Execute str1

Do While Not rs.EOF

Combo1.AddItem (rs.Fields(0))

rs.MoveNext

Loop

rs.Close

con.Close

End Sub

Private Sub update_Click()

''male

If Option1.Value = True Then

Call connect

str1 = "update customer set customer_name= '" & Text1.Text & "',gender='" &
Option1.Caption & "',street= '" & Text2.Text & "',city='" & Text3.Text & "',states='" &
Text4.Text & "',pin_code=" & Text5.Text & ",email_id='" & Text6.Text & "',mobile=" &
Text7.Text & " where customer_id=" & Combo1.Text & " "

con.Execute str1
```

```
MsgBox "Customer Details Updated Successfully...", vbInformation, "Update Customer
Details"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Combo1.Text = "Select customer to update"

con.Close

End If

''female

If Option2.Value = True Then

Call connect

str1 = "update customer set customer_name= '" & Text1.Text & "',gender='" &
Option2.Caption & "',street= '" & Text2.Text & "',city='" & Text3.Text & "',states='"
&Text4.Text & "',pin_code=" & Text5.Text & ",email_id='" & Text6.Text & "',mobile=" &
Text7.Text & " where customer_id=" & Combo1.Text & " "

con.Execute str1

MsgBox "Customer Details Updated Successfully...", vbInformation, "Update Customer
Details"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""
```

```
Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Combo1.Text = "Select customer to update"

con.Close

End If

End Sub
```

SEARCH VEHICLE

```
Private Sub Combo1_click()

Call connect

str1 = "select * from vehicle where vehicle_id=" & Combo1.Text & ""

rs.Open str1, con, adOpenKeyset


Text1.Text = rs.Fields("fuel_type").Value

Combo2.Text = rs.Fields("vehicle_type").Value

Text2.Text = rs.Fields("company_name").Value

Text3.Text = rs.Fields("vehicle_model").Value

DTPicker1.Value = rs.Fields("registration_date").Value

Text4.Text = rs.Fields("vehicle_number").Value
```

```
Text5.Text = rs.Fields("total_seats").Value

Text6.Text = rs.Fields("fuel_capacity").Value

rs.MoveNext

rs.Close

con.Close

End Sub


Private Sub Form_Load()

If con.State = adStateOpen Then

rs.Close

con.Close

End If


With Combo2


    .AddItem "car"

    .AddItem "bus"


 End With

Call connect

str1 = "select * from vehicle"

rs.Open str1, con, adOpenKeyset
```

```
Do While Not rs.EOF

Combo1.AddItem (rs.Fields("vehicle_id").Value)

rs.MoveNext

Loop

rs.Close

con.Close

End Sub


Private Sub cancel_Click()

Unload Me

End Sub


 Private Sub update_Click()

If Combo1.Text = "SELECT VEHICLE ID" Then

MsgBox "Select Vehicle ID to UPDATE", vbExclamation, "Delete vehicle"

Exit Sub

End If


If Combo2.Text = "SELECT VEHICLE TYPE" Then

MsgBox "Please select vehicle type State !", vbExclamation, "Add Vehicle"

Exit Sub

End If
```

```
If Text1.Text = "" Then
MsgBox "Please Enter Fuel Type !", vbExclamation, "Add Vehicle"
Exit Sub
End If

If Text2.Text = "" Then
MsgBox "Please Enter Company Name !", vbExclamation, "Add Vehicle"
Exit Sub
End If

If Text3.Text = "" Then
MsgBox "Please Enter Vehicle Model !", vbExclamation, "Add Vehicle"
Exit Sub
End If
If Text4.Text = "" Then
MsgBox "Please Enter Vehicle Number !", vbExclamation, "Add Vehicle"
Exit Sub
End If

If Text5.Text = "" Then
MsgBox "Please Enter Total Seats !", vbExclamation, "Add vehicle"
Exit Sub
End If
```

```
If Text6.Text = "" Then
MsgBox "Please Enter Fuel Capacity !", vbExclamation, "Add Vehicle"
Exit Sub
End If

Call connect
str1 = "update vehicle set vehicle_type='" & Combo2.Text & "',fuel_type='" & Text1.Text &
"',company_name='" & Text2.Text & "',vehicle_model='" & Text3.Text &
"',registration_date='" & DTPicker1.Value & "',vehicle_number='" & Text4.Text &
"',total_seats=" & Text5.Text & ",fuel_capacity='" & Text6.Text & "'where vehicle_id=" &
Combo1.Text & " "
con.Execute str1
MsgBox "Vehicle details Updated Successfully", vbInformation, "Update vehicle"
con.Close
End Sub

Private Sub delete_Click()
If Combo1.Text = "" Then
MsgBox "Select Vehicle ID to delete", vbExclamation, "Delete vehicle"
Exit Sub
End If
Call connect
confirm = MsgBox("Are you sure you want to delete this vehicle ?", vbYesNo, "Delete a
vehicle")
```

```
If confirm = vbYes Then
str1 = "delete from vehicle where vehicle_id=" & Combo1.Text & ""
con.Execute str1
MsgBox "Vehicle Deleted Successfully...", vbInformation, "Delete vehicle"
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Combo2.Text = ""
Combo1.Text = ""
Else
MsgBox "Vehicle not deleted ", vbInformation, "Delete Vehicle"
End If
con.Close
End Sub
Private Sub clear_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
```

```
Text6.Text = ""

Combo1.Text = ""

Combo2.Text = ""

End Sub
```

UPDATE NOTIFICATION

```
Private Sub Combo1_Click()

Call connect

str1 = "select * from notifications where notification_id=" & Combo1.Text & ""

rs.Open str1, con, adOpenKeyset


Text1.Text = rs.Fields(1)

DTPicker1.Value = rs.Fields(2)

Text3.Text = rs.Fields(3)

Text4.Text = rs.Fields(4)

Combo2.Text = rs.Fields(5)

Text5.Text = rs.Fields(6)


rs.MoveNext

rs.Close

con.Close

End Sub
```

```
Private Sub Form_Load()
If con.State = adStateOpen Then
rs.Close
con.Close
End If

Call connect
str1 = "select * from notifications"
rs.Open str1, con, adOpenKeyset
Do While Not rs.EOF

Combo1.AddItem (rs.Fields(0))
rs.MoveNext
Loop
rs.Close
con.Close
End Sub

Private Sub Image1_Click()
Call connect
str1 = "update notifications set purpose='" & Text1.Text & "' , notification_date='" &
DTPicker1.Value & "',notification_time_H=" & Text3.Text & ",notification_time_M=" &
Text4.Text & ",am_pm='" & Combo2.Text & "',venue='" & Text5.Text & "'where
notification_id=" & Combo1.Text & " "
```

```
con.Execute str1
MsgBox "Notification Updated Successfully", vbInformation, "Update Notification"
con.Close
End Sub

Private Sub Image2_Click()
Unload Me
End Sub
```

DELETE BOOKING

```
Private Sub cancel_Click()
Unload Me
End Sub

Private Sub Combo1_click()
Call connect
str1 = "select * from booking where booking_id=" & Combo1.Text & ""
rs.Open str1, con, adOpenKeyset

Text1.Text = rs.Fields("customer_id").Value
Text2.Text = rs.Fields("customer_name")
```

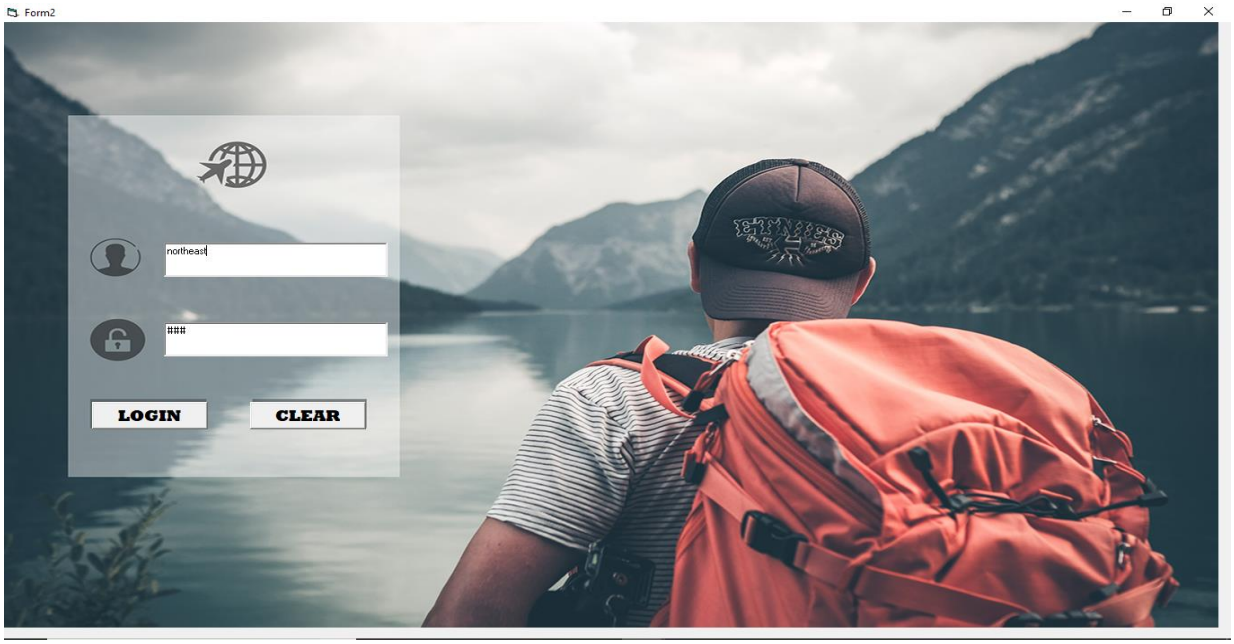
```
Text3.Text = rs.Fields("mobile")

Text4.Text = rs.Fields("state")

Text5.Text = rs.Fields("hotel")

Text6.Text = rs.Fields("number_of_persons")

Text7.Text = rs.Fields("boarding_date")

Text8.Text = rs.Fields("package_id")

Text9.Text = rs.Fields("email_id")

Text10.Text = rs.Fields("spot")

rs.MoveNext

rs.Close

con.Close

End Sub

Private Sub delete_Click()

If Combo1.Text = "SELECT BOOKING ID" Then

MsgBox "Select BOOKING ID to delete", vbExclamation, "Delete Booking"

Exit Sub

End If


Call connect

confirm = MsgBox("Are you sure you want to delete this Booking ?", vbYesNo, "Delete a
Booking")

If confirm = vbYes Then

str1 = "delete from booking where booking_id=" & Combo1.Text & ""
```

```
con.Execute str1

MsgBox "Booking Deleted Successfully...", vbInformation, "Delete a Booking"


Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Text8.Text = ""

Text9.Text = ""

Text10.Text = ""

Combo1.Text = "select another booking id"

Else

MsgBox "Booking not deleted ", vbInformation, "Delete a Booking"

End If

con.Close

End Sub


Private Sub Form_Load()

If con.State = adStateOpen Then

rs.Close
```

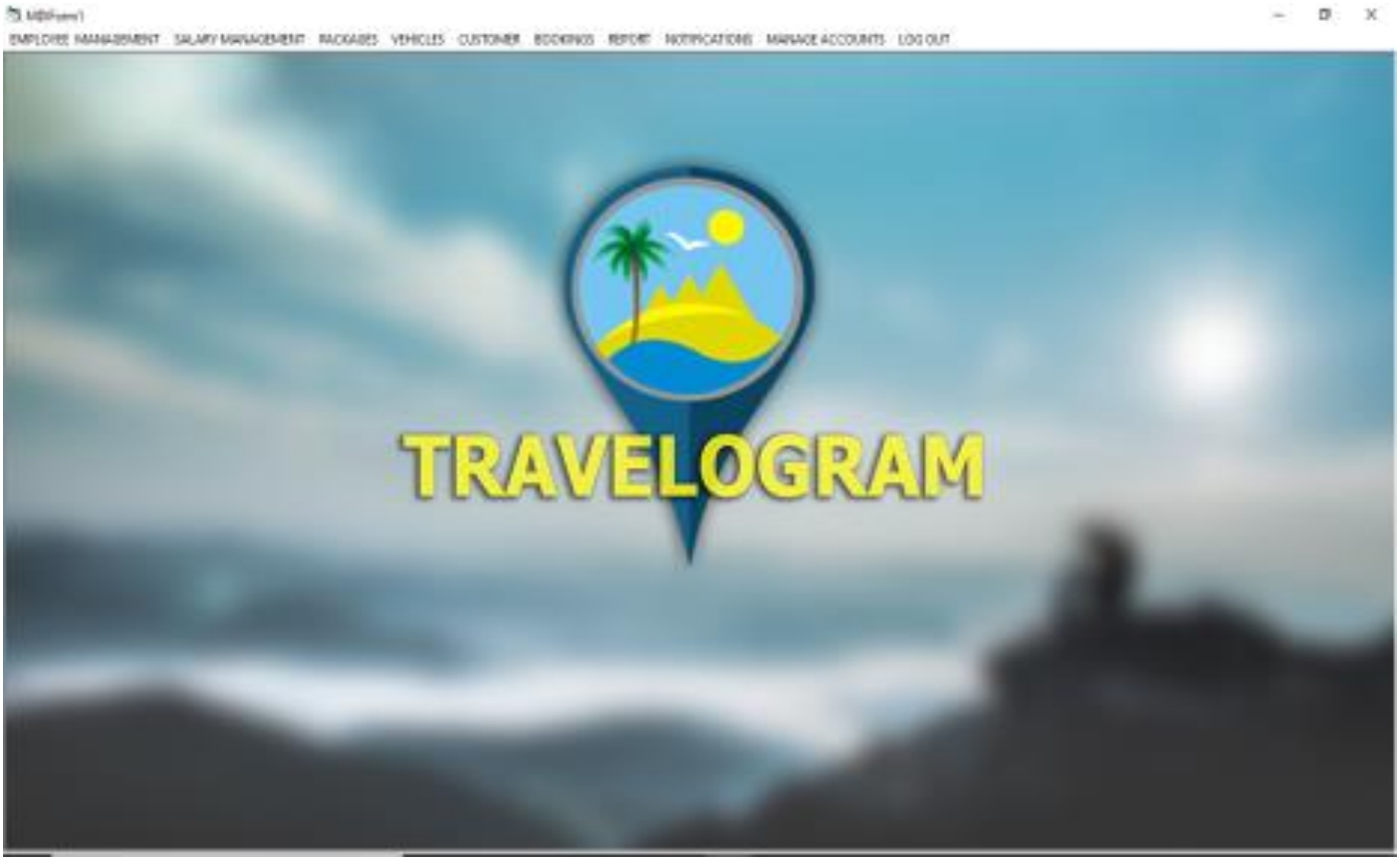
```
con.Close
End If

Call connect
str1 = "select booking_id from booking"
rs.Open str1, con, adOpenKeyset
con.Execute str1
Do While Not rs.EOF
Combo1.AddItem (rs.Fields(0))
rs.MoveNext
Loop
rs.Close
con.Close
End Sub
```

4.2 SCREENSHOTS

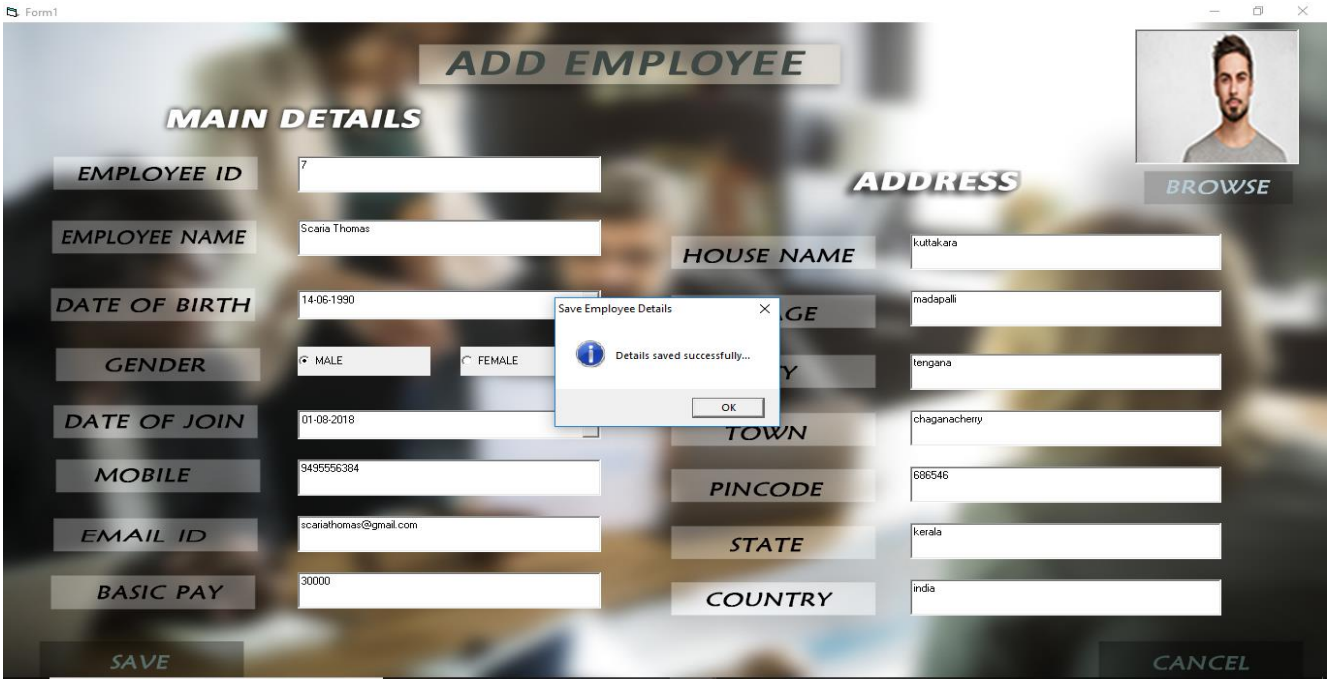
1. Login Form



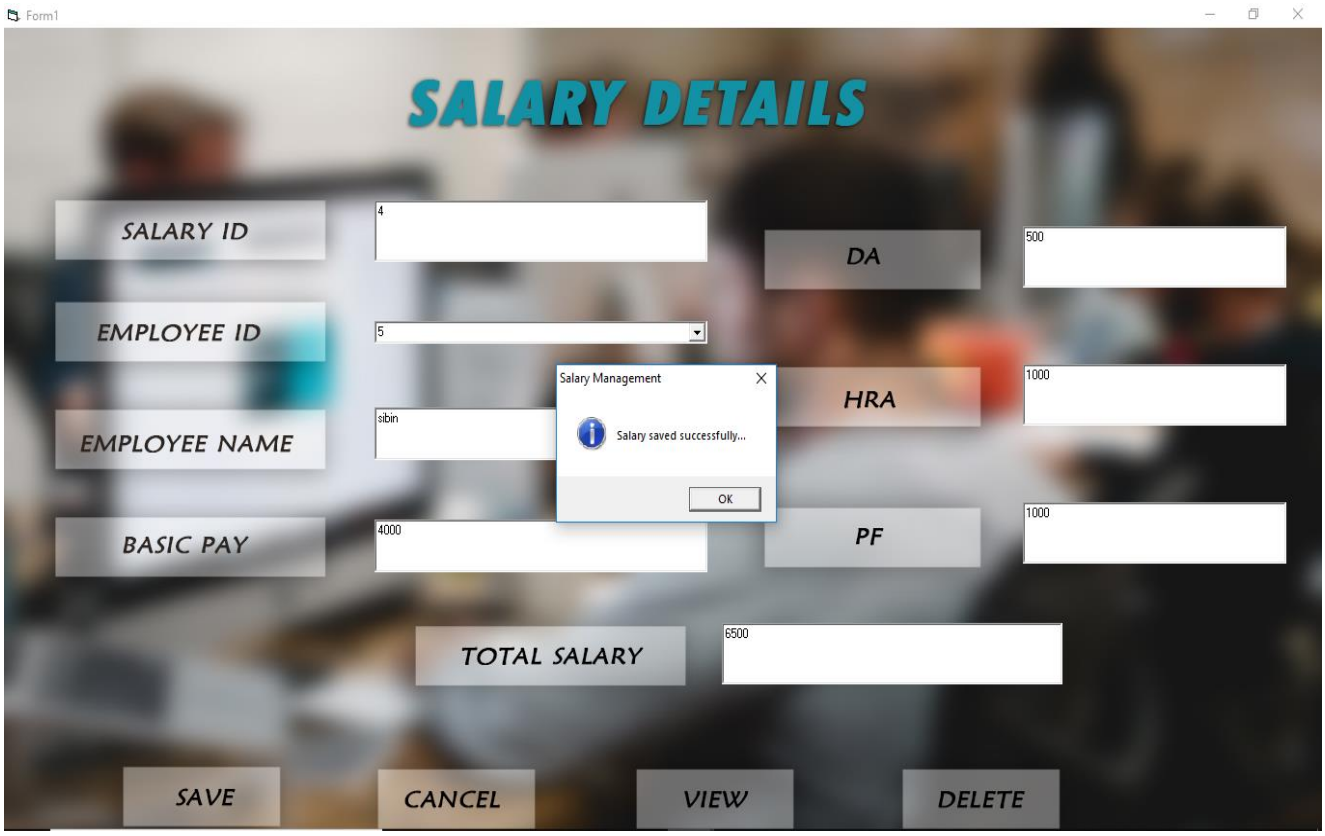
2. Administrator MDI Form



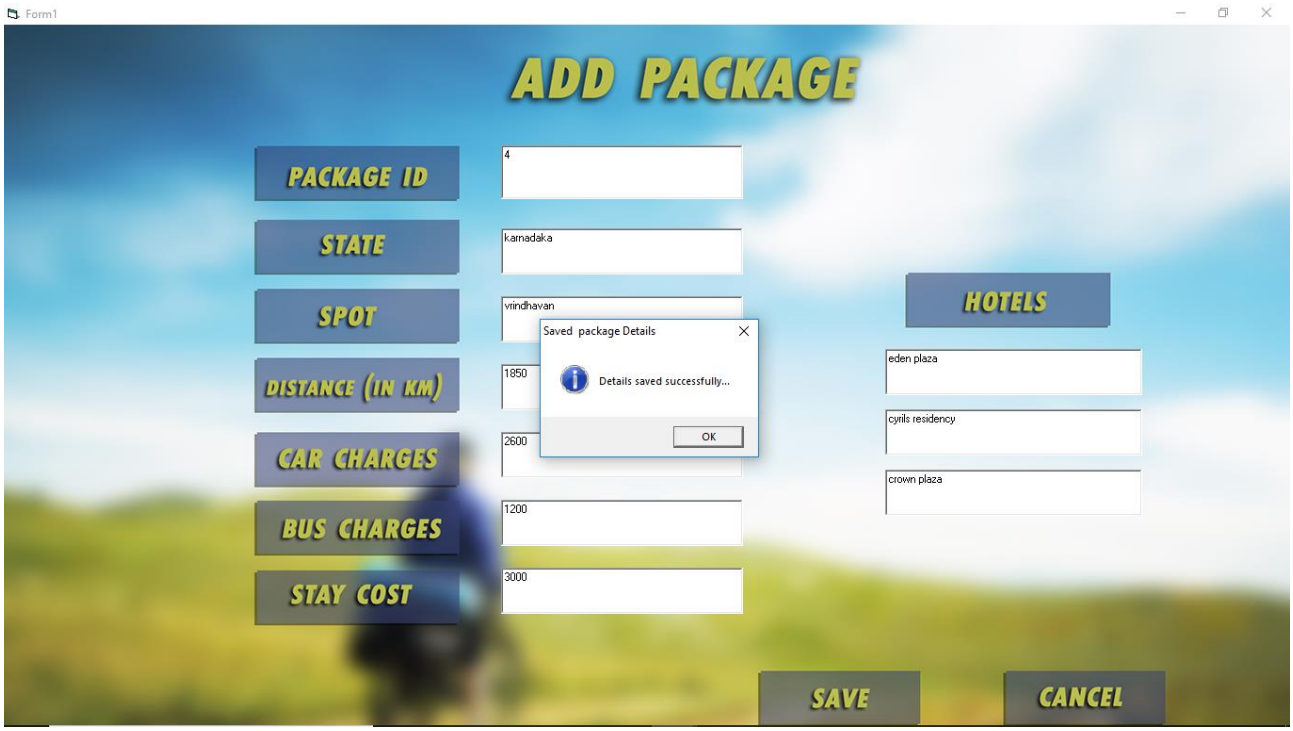
3. Add Employee Form



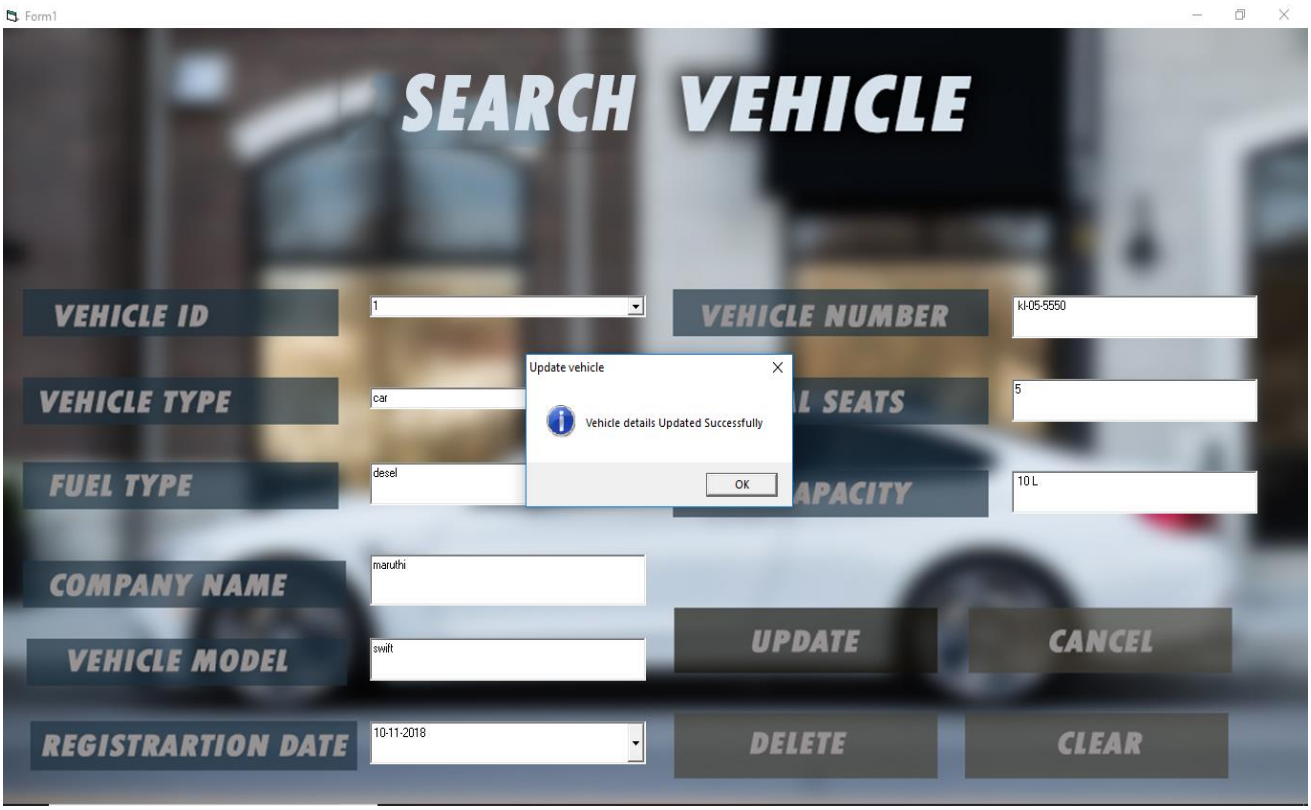
4. Salary Details Form



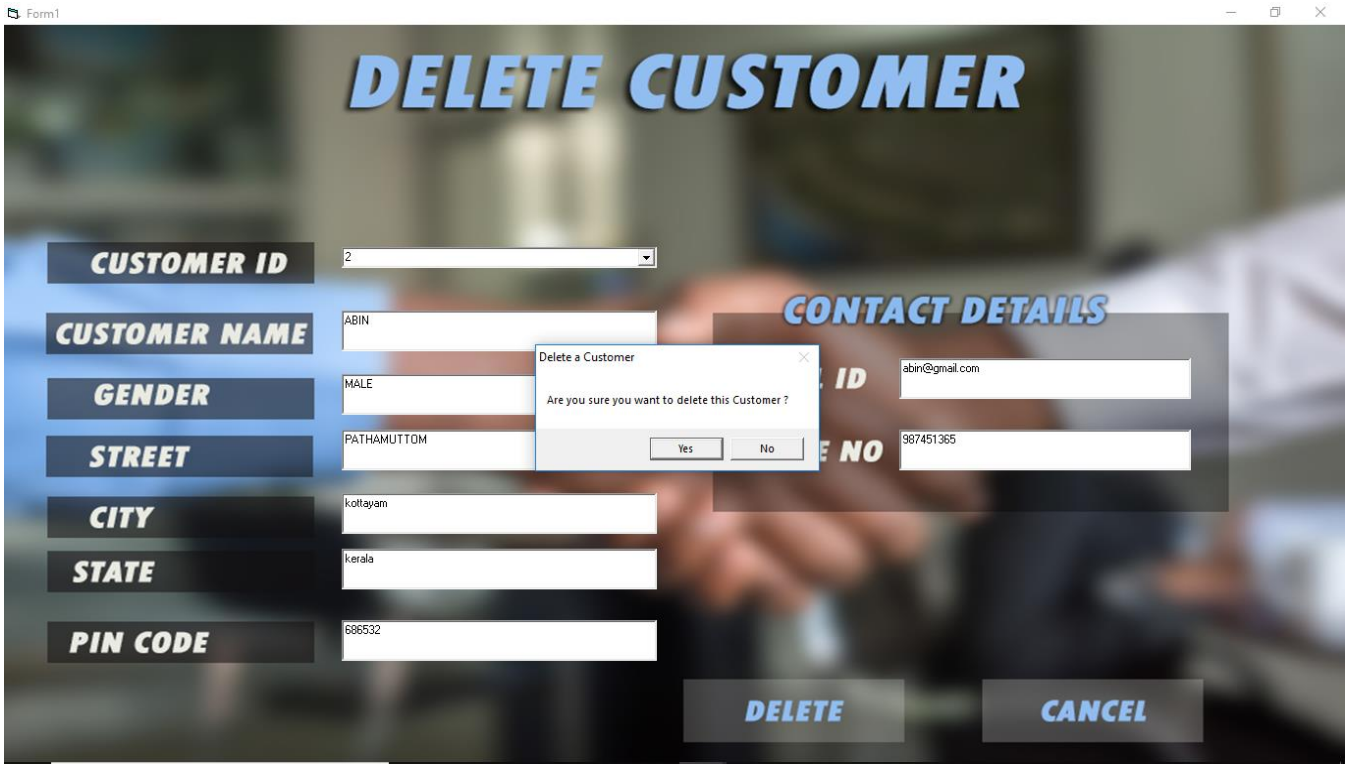
5. Add package Form



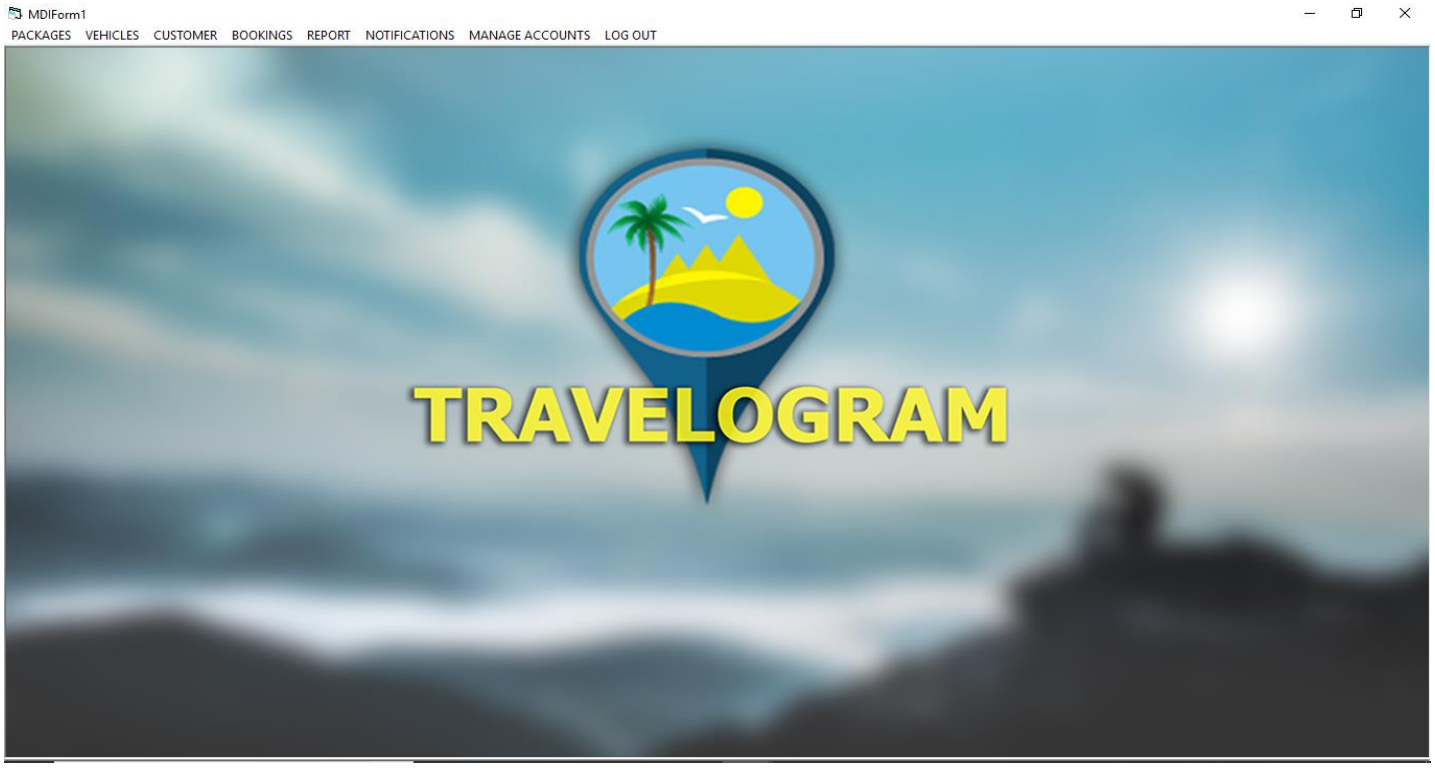
6.Search Vehicle Form



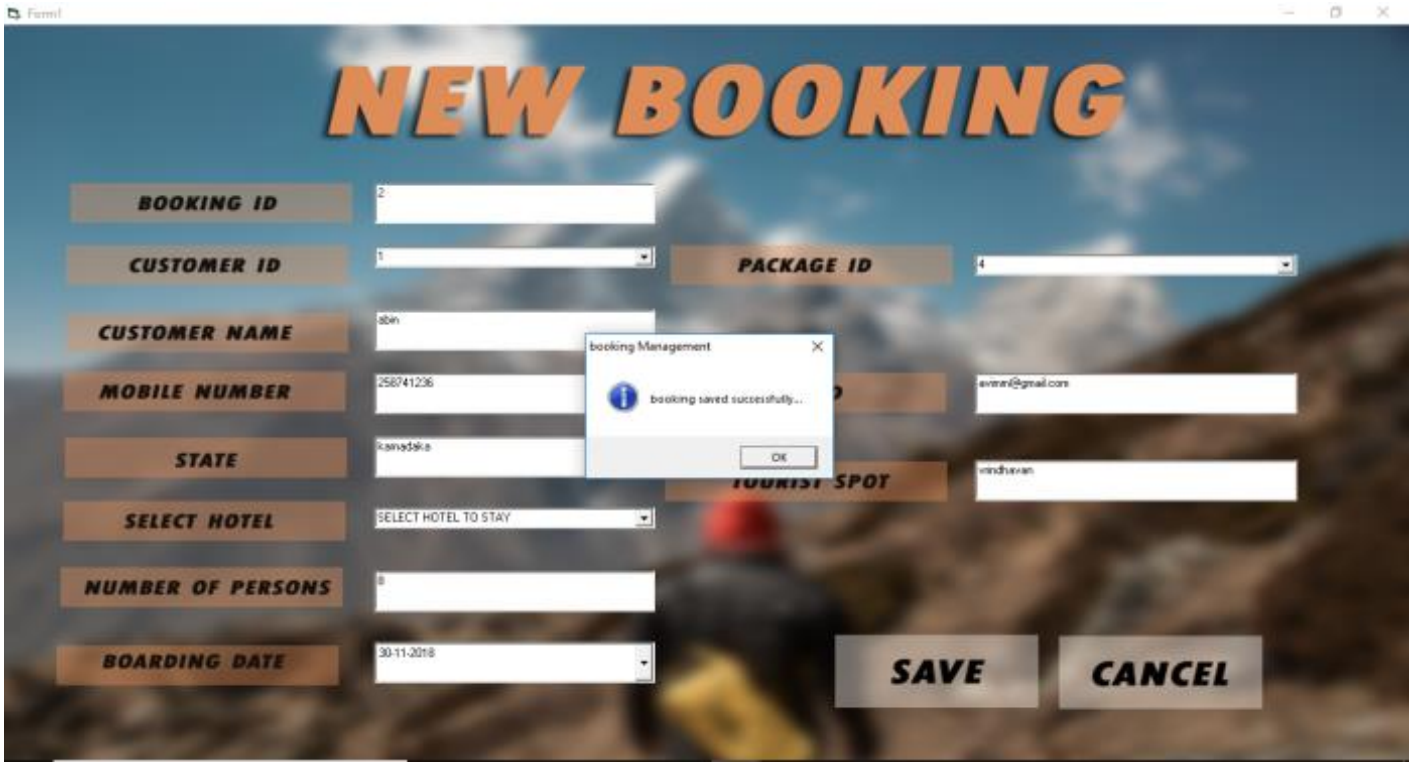
7.Delete customer Form



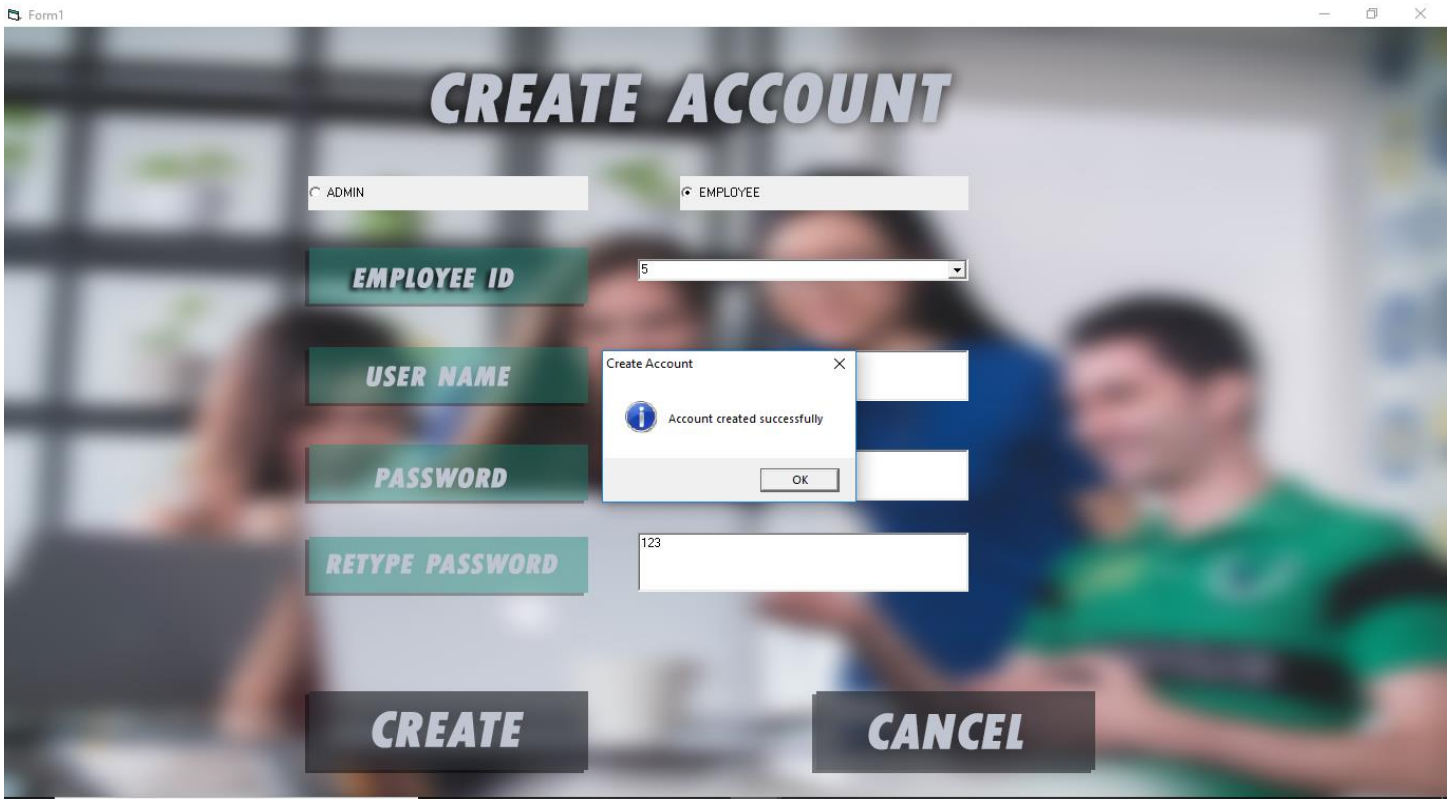
8.Employee MDI Form



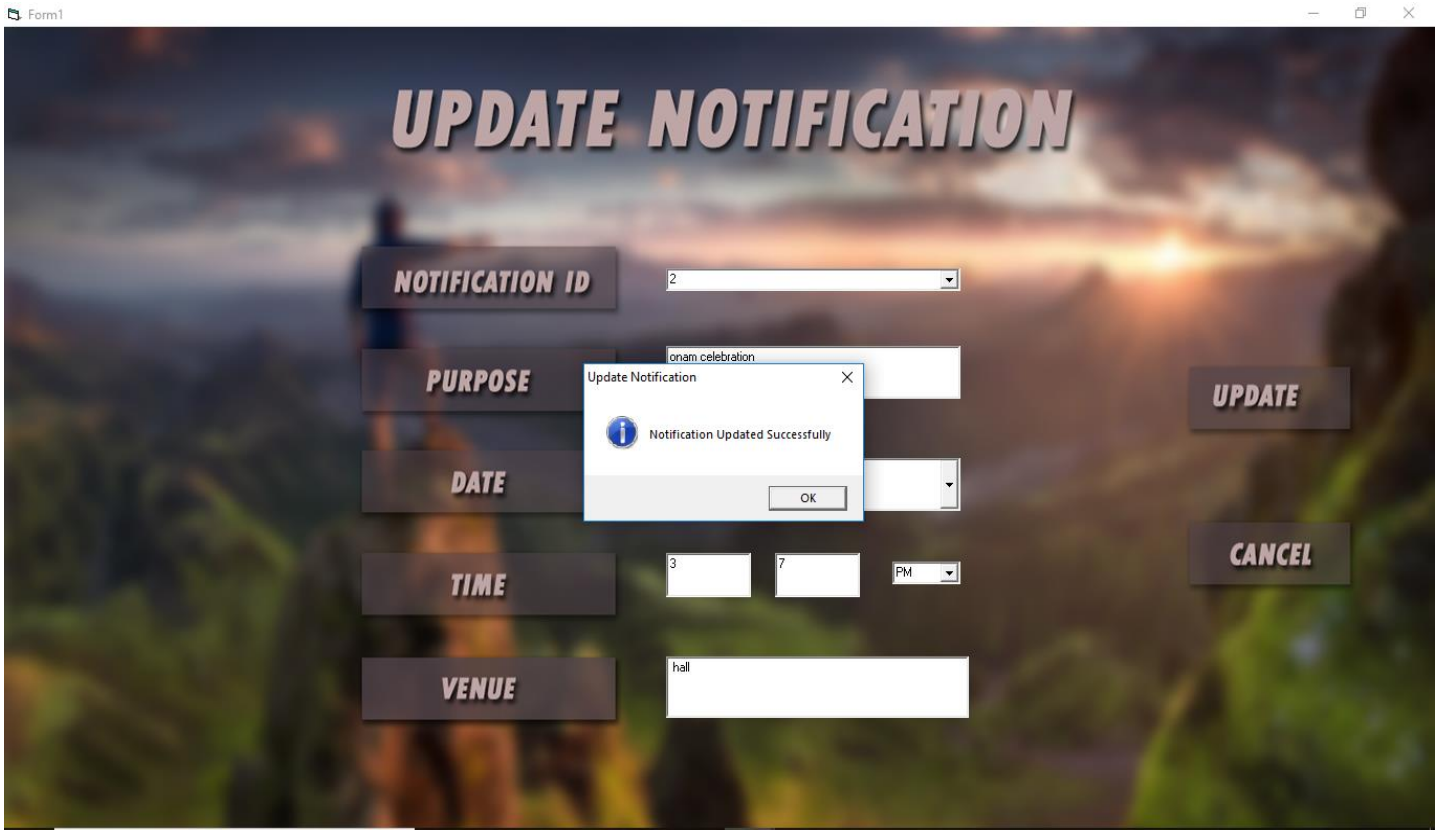
9.New Booking Form



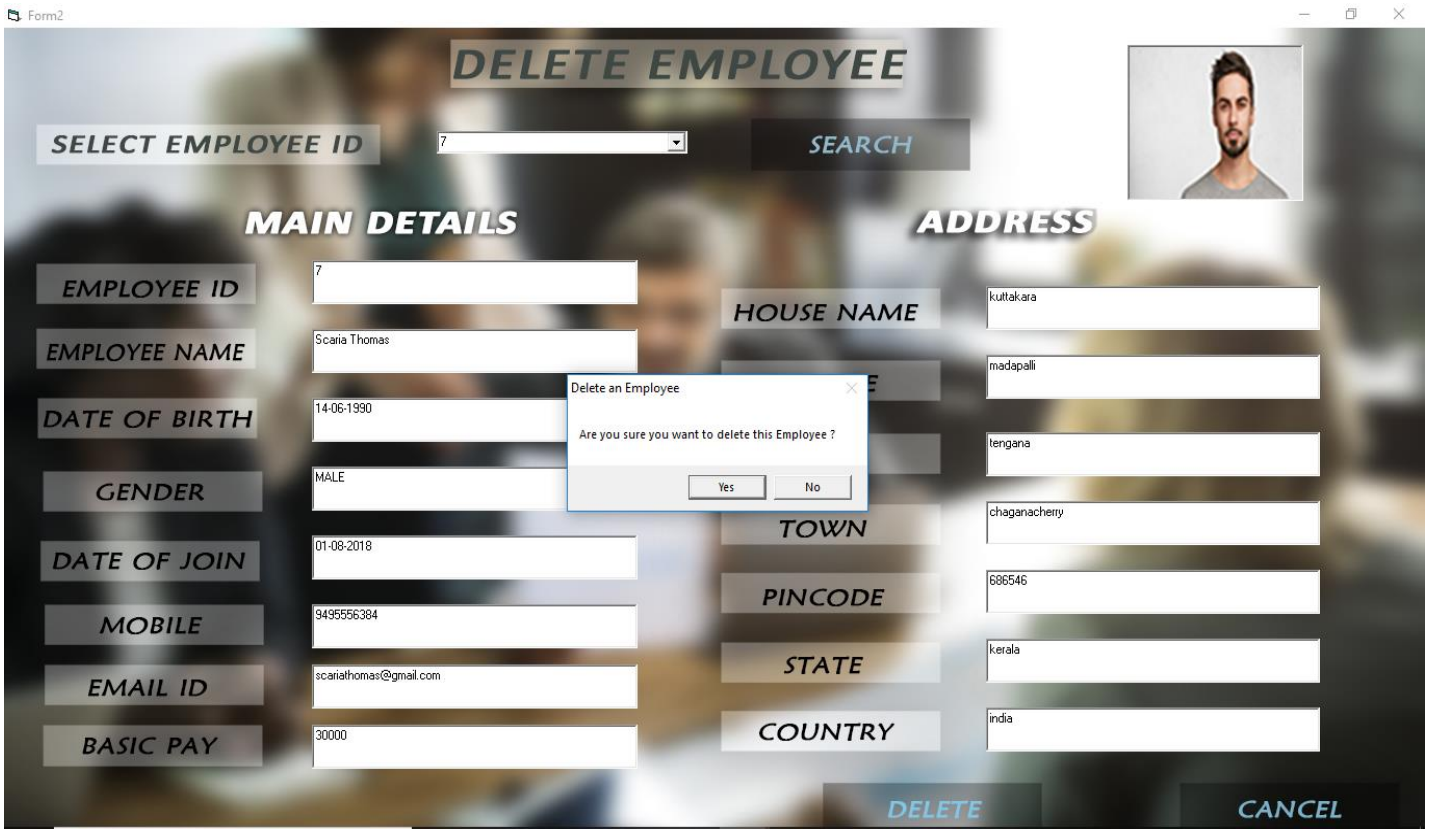
10.Create Account Form



11.Update Notification Form



12.Delete Employee Form



TESTING

5.1 TESTING METHODOLOGY AND STRATEGIES

Software testing is an integral part of ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing. There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time consuming process. Furthermore, it requires skilled and experienced testers to develop great quality test cases. Even with the wide availability of automated tools for testing, the degree of automation mostly remains at the automated test script level, and generally significant amount of human intervention is required in testing. In addition, data collected as testing is conducted provides a good indication of software quality as a while. The debugging process is the most unpredictable part of testing process.

Testing begins at the module level and work towards the integration of the entire computer based system. No testing is complete without verification and validation part. The goals of verification and validation activities are to access and improve the quality of work products generated during the development and modification of the software. Testing pays a vital role in determining the reliability and efficiency of the software and hence is very important stage in software development. Tests are to be conducted on the software to evaluate its performance under a number of conditions. Ideally, it should do so at the level of each module and also when all of them are integrated to form the completed system.

In the project Car show room management system the testing has been successfully handled with the modules. The test data was given to each and every module in all respect has got the desired output. Each module that has been tested is found working properly.

5.2 UNIT TEST REPORT

Here we test each module individually and integrate the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as “Module Testing”. The modules of the Tour management system are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactory as regard to expected output from the module. There

are some validation checks for the fields. Unit testing gives stress on the modules of Tour management system independently of one another, to find errors. Here different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test SQL servers. Programs unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the Tour management system.

5.3 INTEGRATION TEST REPORT

Data can be lost across an interface, one module can have an adverse effect on the other sub functions, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance. In this whole modules of Tour management system are connected and tested.

5.4 SYSTEM TEST REPORT

System testing focuses on testing the system as a whole. It is a crucial step in Quality Management Process. In the Software Development Life Cycle system testing is the first level where the system is tested as a whole. The system is tested to verify if it meets the functional and technical requirements. The application/system is tested in an environment that closely resembles the production environment where the application will be finally deployed. The system testing enables us to test, verify and validate both the business requirements as well as the application architecture.

The prerequisites for system testing are:

- All the components should have been successfully unit tested.
- All the components should have been successfully integrated.

5.5 PERFORMANCE TEST REPORT

Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. Qualitative attributes such as reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing.

PROJECT IMPLEMENTATION

6.1 INTRODUCTION

The implementation is one phase of software development. Implementation is that stage in the project where theoretical design is turned into working system. Implementation involves placing the complete and tested system software into actual work environment. Implementation is concerned with translating design specification with source code. The primary goal of implementation is to write the source code to its specification can easily be verified, and so that debugging, testing and modification can be eased. The goal can be achieved by making the source code clear and straight forward as possible. Implementation means the process of converting a new or revised system design into operational one. The three types of implementation are

- Implementation of a complete system to replace a manual system
- Implementation of a new system to replace existing one
- Implementation of a modified application to replace an existing one.

6.2 IMPLEMENTATION DETAILS

The process model used in this project is “waterfall model”.

The implementation for the process model as follows:

1. **Requirement analysis and project planning**

The project started on 30-7-2018 and the first phase was to analyse the requirements, like who will be the end user and what was total expected duration for completion of project. There were several meeting conducted with project-in-charge.

2. **For project planning there were two modules.**

Admin module

User module

1.Design

The second phase of the implementation was design phase. In this phase before the coding the diagrams were designed such as

- DFD

2.Coding

The coding phase is the next phase in the model. The coding was done with the VISUAL BASIC 6.0 which handled the front end and MS SQL Server 2008 handles the backend.

3.Testing

Once the coding phase gets completed the next major phase was the testing phase. System testing was conducted against the requirements and 90% of the requirements got satisfied.

4.Implementation

The final phase of the project is the implementation. The project was implemented and worked correctly in the local host.

FRONTEND DESCRIPTION:

MICROSOFT VISUAL BASIC 6.0

Visual Basic is a third-generation event-driven programming language and integrated development environment (IDE) from Microsoft for its COM programming language, because of its graphical development features and BASIC heritage.

Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects. Scripting languages such as VBA and VB Script are syntactically similar to Visual Basic, but perform differently.

A programmer can create an application using the components provided by the Visual Basic program itself. Programs written in Visual Basic can also use the Windows API, which requires external functions declarations.

The final release was version 6 in 1998. Microsoft's extended support ended in March 2008 and the designated successor was Visual basic .NET (now known simply as Visual Basic).

HISTORY

VB 1.0 was introduced in 1991. The drag and drop design for creating the user interface is derived from a prototype form generator developed by Alan Cooper and his company called Tripod. Microsoft contracted with Cooper and his associates to develop Tripod into a programmable form system for Windows 3.0 under the code name Ruby.

Tripod did not include a programming language at all. Microsoft decided to combine Ruby with the Basic language to create Visual basic.

The ruby interface generator provided the "visual" part of visual basic and this was combined with the "EB" Embedded BASIC engine designed for Microsoft's abandoned "Omega" database system. Ruby also provided the ability to load dynamic link libraries containing additional controls (then called "gizmos"), which later become the VBX interface.

VB VERSIONS

Visual Basic is available in three versions, each geared to meet a specific set of development requirements.

- **Learning Edition**

The visual basic learning edition allows programmers to easily create powerful applications for Microsoft Windows 95 and Windows NT. It includes all intrinsic controls, plus grid, tab, and data-bound controls.

Documentation provided with this edition includes learn VB Now, a printed Programmer's Guide, online Help, plus Visual Basic Books Online.

- **Professional Edition**

It provides computer professionals with a full-featured set of tools for developing solutions for others. It includes all the features of the Learning edition, plus additional ActiveX controls, including Internet controls, and the crystal report writer. Documentation provided with the professional edition includes the programmer's guide, online help, the component tools guide and the crystal reports for visual basic user's manual. The professional edition includes all of the data control features, and in the addition provides the full data access objects programming interface.

- **Enterprise Edition**

This allows professional to create robust distributed applications in a team setting. It includes all the features of the professional edition, plus the automation manager, component manager, database management tools, the Microsoft Visual SourceSafe ™ project-oriented version control system, and more. Printed documentation provided with the Enterprise edition includes all Professional edition pieces, plus the Building Client/server applications with visual basic book and the SourceSafe user's guide.

6.3 LANGUAGE FEATURES

Like the BASIC programming language, Visual Basic was designed to accommodate a steep learning curve. Programmers can create both simple and complex GUI applications. Programming in VB is a combination of visually arranging components, and writing additional lines of code for more functionally. Since VB defines default attributes and actions for the components, a programmer can develop a simple program without writing much code. Programs built with earlier versions suffered performance problems, but faster computers and native code complication has made this less of an issue.

Although programs can be complied into native code executables from version 5 onwards, they still require the presence of around 1MB of runtime libraries. Runtime libraries are included by default in windows 2000 and later. Earlier versions of windows (95/98/NT), require that the runtime libraries be distributed with the executable.

Forms are created using drag- and – drop technologies. A tool is used to place the controls (examples: textboxes, buttons etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attributes values can be modified during runtime based on user actions or changes in environment, providing a dynamic application. For example, code can be inserted into form resize event handler to reposition a control so that it remains centred on the form, expands to fill up the form, etc. by inserting code into the event handler for a key press in a textbox, the program can automatically translate in the case of text being entered, or even certain characters from being inserted.

Visual Basic can create executable (EXE files), ActiveX controls, or DLL files, but is primarily used to develop windows applications and to interface database systems. Dialog boxes with less functionality of the application, while programmers can insert additional logic within the appropriate event handlers.

For example, a drop-down combination box will automatically display its list and allow the user to select any element. An event handler is called when an item is selected, which can then execute additional code that the programme created to perform the action for that list item.

Alternatively, a visual basic component can have no user interface, and instead provide ActiveX objects to other programs via Component Object Model (COM). This allows for server-side processing or an add-in module.

The language is garbage collected using reference counting, has a large library of utility objects, and has basic object oriented support. Unlike many other programming languages, visual basic is generally not case sensitive-though it transforms keywords into a standard case configuration and forces the case of variable names to conform to the case of the entry in the symbol table. String comparisons are case sensitive by default.

The visual basic compiler is shared with other visual studio languages (C,C++), but restrictions in the IDE do not allow creation of some targets(windows model DLLs) and threading models.

ADVANTAGES

- The structure of the basic programming language is very simple, particularly as to the executable code.
- VB is not only a language but primarily and integrated, interactive development environment (“IDE”).
- The VB – IDE has been highly optimized to support rapid application development (RAD). It is particularly is to develop graphical user interface and to connect them to handler functions provided by the application.
- The graphical user interface of the VB-IDE provides intuitively appealing views for the management of the program structure in the large and the various types of entities (classes, modules, procedures, forms etc.).
- VB provides a comprehensive interactive and context- sensitive online help system.

- When editing program texts the" IntelliSense" technology informs you in a little pop up window about the types of constructs that may be entered at current cursor location.

DISADVANTAGES

- Versioning problems associated with various runtime DLL's.
- Poor support for Object- Oriented programming.
- Inability to create multi-thread applications, without resorting to windows API calls.
- Lack of Unicode support

BACKEND DESCRIPTION

MICROSOFT SQL SERVER MANAGEMENT STUDIO

Microsoft SQL Server 2008 is a set of components that work together to meet the data storage and analysis needs of the largest Websites and enterprise data processing systems. SQL server is a relational database management system for distributed client-server computing. It provides the following features

- A variety of user interfaces
- Physical data independence
- Logical data independence
- Query optimization
- Data integrity
- Concurrency control
- Backup & recovery
- Security and authorization

QUERY ANALYSER

Query analyser is another tool with SQL server, which extends the capabilities of ANSI standard SQL. It is an application that recognizes and executes SQL commands and specialized T_SQL commands that can be used to create database objects using SQL commands. We can use query analyser commands to

- Enter, edit, store and retrieve and run SQL commands
- Format, perform calculations on, store and print query results

- List column definitions for any table
- Access any copy of data between SQL databases
- We can create tables and insert data or alter it or we can delete data using SQL commands

FEATURES OF SQL SERVER 2008

Microsoft SQL server features include:

- Internet integrations
- The SQL server 2008 database engine can be used across platforms ranging from laptop computers running Microsoft windows 98 through large, multiprocessor servers running Microsoft windows 2008 Data Sender Edition.
- Scalability and availability
- SQL server 2008 database engine includes integrated XM support.
- Enterprise- level database features
- The SQL server 2008 relational database engine supports the features required to support demanding data processing environments.
- Ease of installation, deployment and use.
- SQL server 2008 includes a set of administrative and deployment tools that improve upon the process of installing, deploying, managing and using SQL server across several systems.
- Data Warehousing.
- SQL server 2008 includes tools for extracting and analysing summary data for online analytical processing.

ADVANTAGES OF SQL SERVER 2008 AS A DATABASE SERVER

Microsoft SQL server 2008 is capable of supplying database services needed by extremely large systems. Large servers may have thousands of user's connector to an instance of SQL server 2008 at the same time. SQL server 2008 has full protection for these environments, with safeguards that prevent problems, such as having multiple users trying to update the same piece of data at the same time. SQL server 2008 also allocates the available resources effectively, such as memory, network bandwidth, and disk I/O, among the multiple users.

CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

The project "**TOUR MANGEMENT SYSTEM"** is a dynamic application that allows each user to access. It has been designed in such a way that it is easy to modify and can be updated efficiently. The forms are designed in a user friendly manner by providing messages and captions so that any user friendly manner by providing messages and captions so that any user can this system most efficiently. This application was tested with real data and was found to be error free and the system will work successfully. All the necessary validations are carried out in this project.

The major advantages are:

- Easy retrieval of data available in database.
- Very user friendly.
- Reduces manual work.
- Very cost effective.

7.2 FUTURE SCOPE OF THE PROJECT

As all the projects have their future expansion. The entire project may be changed in future by the other person. In the same way we also have our project future expansion. In future somebody might buy this project. Also some might add some additional features to these projects and can expand these projects. But these are not yet created in any of the college so we decided to make these projects. Also in future some industries might be interested in buying this project. And also we will be in future we will try to make these projects more interesting by adding some features in it. Also this is developed in visual basic language but in future it can be developed in some other language but it is very easy to develop in this language as it is user friendly. Also in this we have provided the ado connection so somebody in future if expanding the project then he might use some other connectivity.

BIBILOGRAPHY

8.1 REFERENCES

BOOKS

- Norton- “Guide to VISUAL BASIC 6.0”
- Henry Kerb- “DATA BASE Design Concept”

WEBSITES

- www.w3school.com
- www.codeproject.com