PROJECT REPORT ON
Tours and Travels Management System

SUBMITTED TO
UNIVERSITY OF MUMBAI

BY

Sreeraj Menon

GUIDED BY
PROF. Binita Thakkar

T.Y.B.Sc. (INFORMATION TECHNOLOGY)
YEAR 2016-2017

LATE SHRI VISHNU WAMAN THAKUR CHARITABLE TRUST'S
BHASKAR WAMAN THAKUR COLLEGE OF SCIENCE
YASHWANT KESHAV COLLEGE OF COMMERCE
VIDYA DAYANAND PATIL COLLEGE OF ARTS
VIRAR (W), TAL. VASAI, DIST. THANE 401303

Late Shri Vishnu Waman Thakur Charitable Trust's
Bhaskar Waman Thakur College of Science,
Yashwant Keshav Patil College of Commerce,
Vidya Dayanand Patil College of Arts.
Virar (W).

CERTIFICATE

This is to certify that project done on _____by Mr. _____Seat no._____ in partial fulfillment of B.Sc.IT degree (SEM VI) examination had not been submitted for any other examination and does not form any other course undergone by the guide.

_____ _____ _____

Project Guide **Examiner** **Head of Dept.**

Date: **Date:** **Date:**

COLLEGE SEAL:

PREFACE

It gives me great pleasure to present the project on "**Tours and Travels Management System"** prepared with my fullest sincere efforts.

The content of this book are presented into many chapters, in order to ease reading. The project has been illustrated with precise data with State Chart Diagram, Use Case Diagram, Class Diagram that helps in understanding of the software. This project includes the software development tools like, Netbeans IDE to present the software very precisely.

ACKNOWLEDGEMENT

I started this project as a part of my course curriculum. It gives me great pleasure to present the report of this project work conducted towards the fulfillment of the project titled " **Tours and Travels Management System** " assigned to me by "**Michael Tours and Travels**".

I take this opportunity to thank those who have made the efforts in success of the project. I extend my special gratitude towards **Prof. Sampada Deshmukh** Head of the Department of Information Technology and Computer Science and lecturer **Prof. Binita Thakkar** (internal guide) who has been a constant source of motivation, encouragement, and guidance that has gone a long way in helping the completion of this project. I express my warm wishes to the entire staff members for their assistance and kind guidance who helped me out of all my queries.

I also heartily thankful to all my friends and "Live Life" for providing me all the useful requirements that were needed by me for the completion of project.

-Sreeraj Menon

INDEX

Sr. No.	Topic
1.	Introduction
1.1	Organization Profile
1.2	Introduction to Project
2	System Study and Analysis
2.1	Existing System
2.2	Proposed System
3.	Planning Phase
3.1	Feasibility Study
3.2	Milestone
3.3	Fact finding Techniques
4.	Front-end And Back-End
5.	Hardware & Software Requirements
6.	System Design/Development
6.1	Gantt Chart
6.2	E-R Diagram
6.3	Event Table
6.4	UML Diagram
6.4.1	Class Diagram
6.4.2	Use-Case Diagram
6.4.3	Sequence Diagram
6.4.4	State Chart Diagram
6.4.5	Activity Diagram
6.4.6	Menu Tree
6.4.7	Database Design
6.4.8	CRUD Table
7.	System Coding Convention
8.	Screen layout with test data & test result testing
9.	Source Code
10.	Report screen layouts
11.	Report Source Code
12.	Test Cases
13.	Future Enhancement
14.	Conclusion
15.	Bibliography, References, Websites

INTRODUCTION

1.1 ORGANIZATION PROFILE

Name	:- Michael Travel Service
Address	:- A/104, Lawrence Shopping Centre, Catholic Bank Building, Vasai Road (west).
Telephone No	:- 2347034,2339970,2335862 Mobile No.9967913147.
Manager	:- Mr. Michael

1.2 ORGANIZATIONAL HIERARCHY

Michael Travel Service would like to introduce itself as one of the emerging Tours and Travels Agency in Vasai keeping the most important factors of travel Safety, Security and quality services for its customers. This organization was founded by Mr.Michael on 1986.

The guiding principal of safe travel has always been the key factor for its customers / clients to extend their excellent service.

They start with Safety and Security, Discipline, Sincerity and integrity with Quality and Punctuality in a professional manner keeping its customers centric with innovation, moving towards Social and Environmental responsibility with a passion of customer service.

The mission of Michael Tours and Travels is to satisfy and gratify the customers by providing them the finest & Economical tour packages as per their requirements with utmost comfort.

Michael Travel Service Highlights:

- 24 hour round the clock service.
- 365 days working.
- Centrally located office in Vasai (Mumbai)
- Follow up with Clients for services rendered.

SYSTEM STUDY AND ANALYSIS

2.1. Existing System:

- All Work are done Manually.

- In Manual Booking System Customer has to go to the Tours and Travel Agency.
- Ask Inquiry for Travelling then Book ticket Finally Pay Payment & Collect Receipt.
- Difficult To Maintain the Customer Details of Package and Payment
- Receipt in Register.
- They Register Tour Package in the notebook.
- Add advertisement in Local newspaper or Local Market.
- Use Travelling Facility For the Limited Area or Person.

2.2. Proposed System:

- To Create Web Based Application For our Organization.
- To Provide Search Facility For Customer.
- To Generate Different Types of Reports.
- To Provide the online Package Ticket Booking and online Payment Facility
- For Customer.
- To Provide package Details.
- Customer Can Cancel the Booking then Return 15% Less Deduct
- From the Amount.
- Services provided by Tour and travels System -
    - **VIEW PACKAGE**
    - **SEARCH PACKAGE**
    - **BOOKING**
- **CANCEL BOOKING**
- **ONLINE PAYMENT**

PLANNING PHASE

3.1 FEASIBILITY STUDY:

The feasibility study is undertaken to determine the possibility of either improving the present system or developing a completely new system. It helps to obtain an overview of the problem and get an idea whether a feasible solution exists.

- **Operational Feasibility Study :**

The current system is manual. Thus processing large amount of data becomes a cumbersome activity. Reports that are generated are difficult to prepare manually and are also error prone. Operationally the Proposed system is feasibility because there are sufficient supports for project from management. It will be running smoother and faster that of existing system.

- **Technical Feasibility Study:**

Generally, new system brings new technology into an organization. The proposed system requires technology and equipment, which can be obtained. the operating system has the technical capacity to hold the data required to use the proposed system. The present equipment technology assures technical guarantee of accuracy, reliability and ease of access.

- **Legal/Ethical Feasibility Study:**

Determines whether the proposed system conflicts with legal requirements.

- **Schedule Feasibility Study:**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable.

- **Resource Feasibility Study:**

This involves questions such as how much time is available to built the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies.

- **Cultural Feasibility Study:**

In this stage, the project's alternatives are evaluated for their impact on the local and generated culture. For example, environmental factors need to be considered and these factors are to be well known. Further an enterprise's own culture can clash with the results of the project.

3.2 MILESTONES:

Sr. No	Activity	Description	Period
1.	Planning	In this phase I have plan out what the modules of my project & how to make those modules.	3 weeks
2.	Requirement	In this phase I have noted down all requirement for my project.	4 weeks
3.	Analysis	In this phase I have analyze the old system & solve those limitation into my software.	5 weeks
4.	Database Design	In this phase I have designed my database tables for my project.	2 weeks
5.	Form Design & Report Design	In this phase I have designed all the project forms & project reports.	2 weeks
6.	Coding	In this phase I have started my hard coding of my project	7 week
7.	Test & Implementation	After my coding phase is completed I started with testing of my software. After software works successful I have implement that software on the system of an organization.	3 week

3.3 Fact Finding Techniques:

We mainly used three fact finding techniques to find out for ourselves the correct information on the basis of which we will build the software.

These fact finding techniques are extremely important because these are the facts on basis of which we can build the software that comprises of a friendly environment for the members work with. This is the reason why fact finding is an important activity grouped under the second phase “Requirement Analysis” of the Software Development life cycle.

The techniques which were used by us include:

- **Interview and Questionnaire**
- **Document sampling**
- **Work site observation**

Interview & Questionnaire:

This is considered as one of the best fact finding technique. This include direct interaction with the customer. It is considered as the best technique, because it is the only way the user can reveal the details & fact about his past, present & expected working, requirements, technologies .These are the information which give us the description of the systems, we have to we have to implement our logic and our own ideas & make this description to turn to reality, to work as module which the user desire.

Questionnaire:

Open-Ended Questions:

Q.1) How does your current process work?
Ans: All the information about employee are stored in the files similarly the customer details are stored in customer file.

Q.2) What are the main modules and attributes of your current System?
Ans: New register details, complaint details, payment, booking and report are the main modules etc. are the attributes.

Q.3) What are limitation of current process?
Ans: Functionality it's a complex procedure, technically it's difficult, lengthy and time consuming.

Close-Ended Questions:

Q.1) What is your Organization name?
Ans: Michael Tours and Travels.

Q.2) Do the Organization have any branches?
Ans: No.

Q.3) On which platform your work manual report is prepared?
Ans: MS-Excel, MS-Word.

Q.4) Is your current system economically, functionally and technically feasible?
Ans: No.
Q.5) How many days it takes to submit my new proposed software?
Ans: 10-15 days.

FRONT-END

AND

BACK-END

4. Front-end and Back-end:

Front-end: Java, Bootstrap (Framework)
Back-end: Mysql

5. Hardware and Software Requirements:

Minimum Hardware Requirements:

Processor: PIII 500MHZ or above
RAM: 128MB RAM
Hard Disk: 100MB Free Hard disk space
Monitor: Standard Color Monitor

Minimum Software Requirements:

Operating System: Any Windows Family
Software : Java
Database: MySQL

SYSTEM DESIGN AND DEVELOPMENT

6. SYSTEM DESIGN & DEVELOPMENT

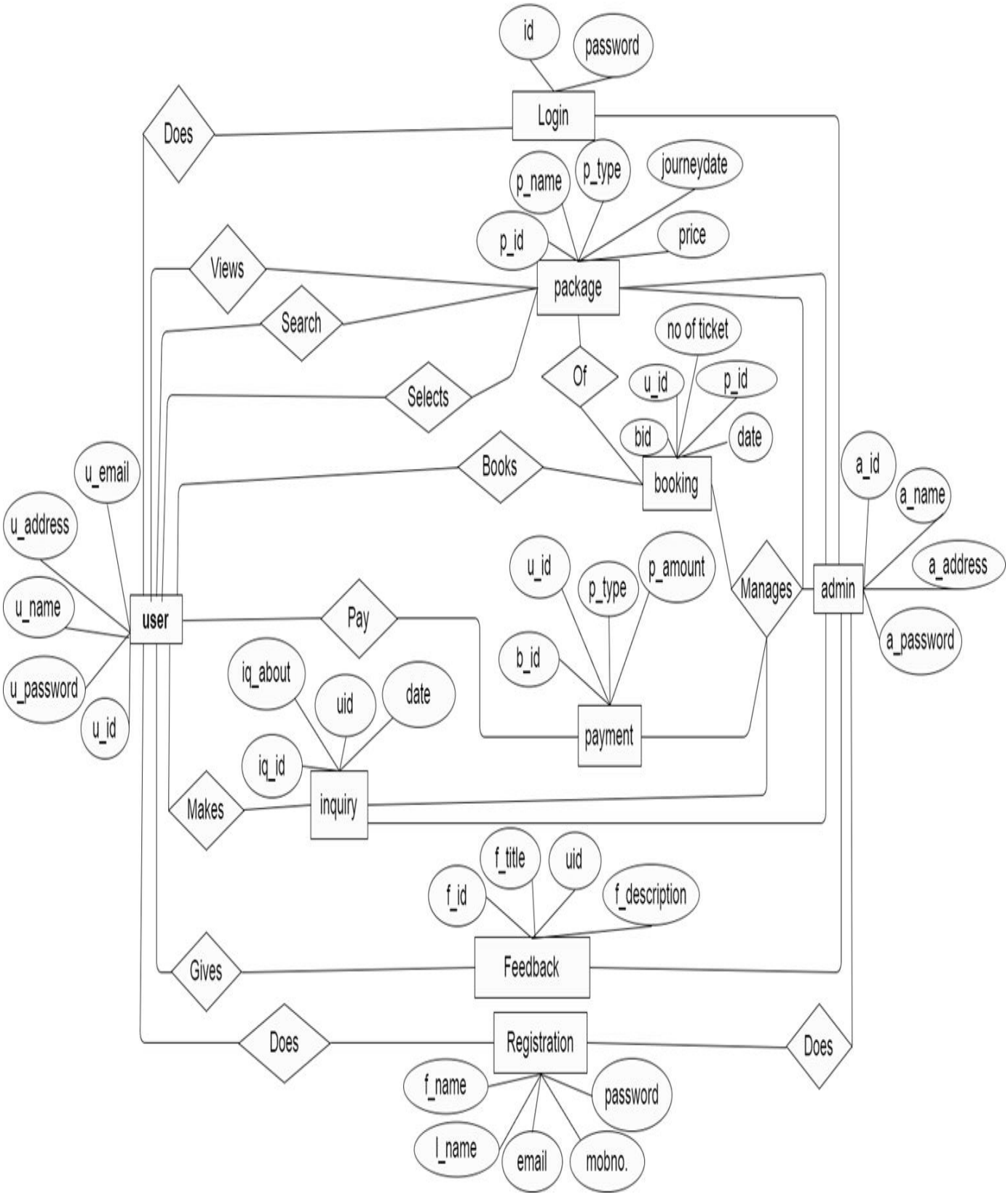
- System design is the solution to the creation of a new system. This phase is composed of several systems. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design.

- During logical design phase the analyst describes inputs (sources), out puts (destinations), databases (data sores) and procedures (data flows) all in a format that meats the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design.

- The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen

6.1 GANTT CHART

6.2ER DIAGRAM



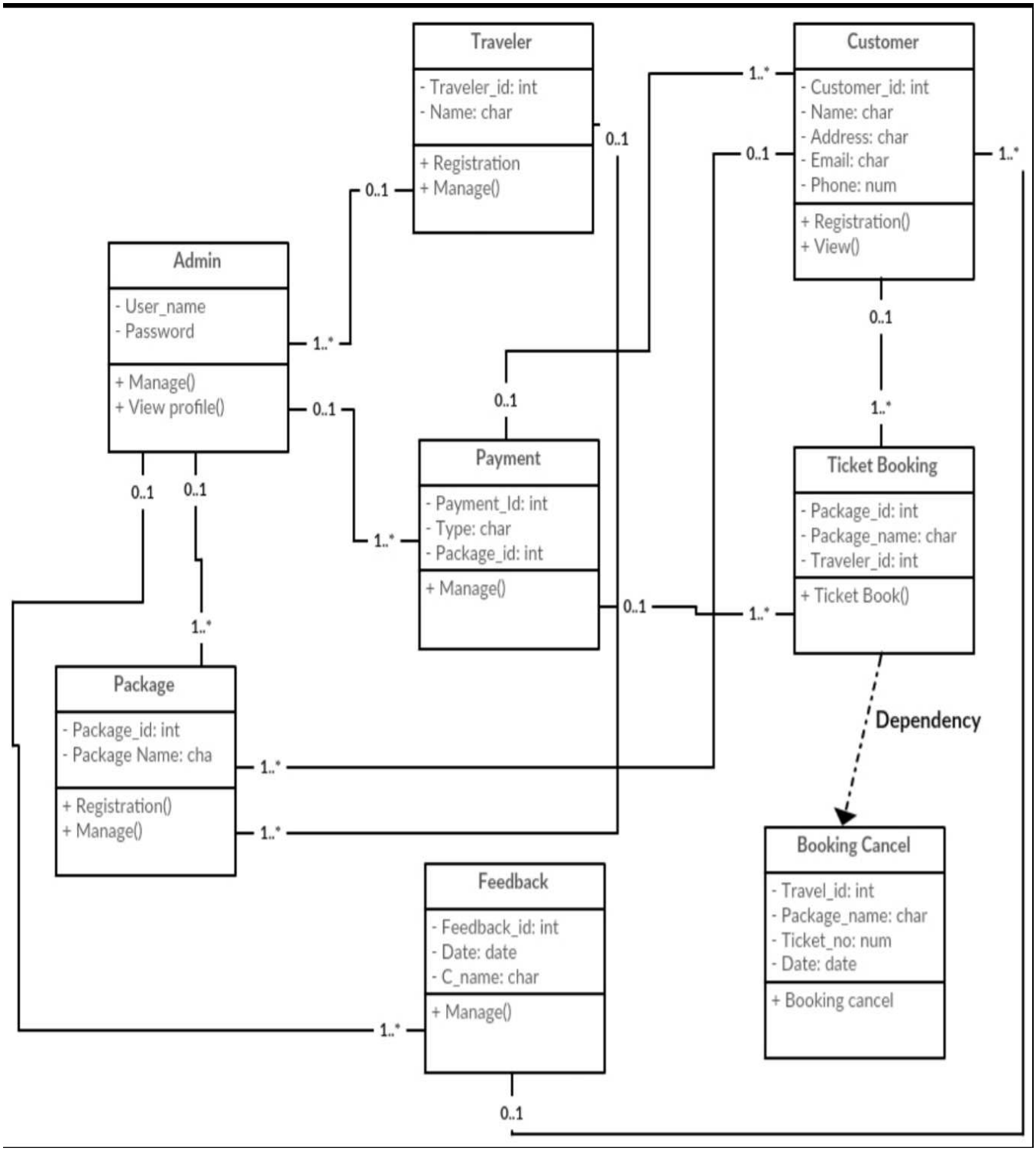
6.3 EVENT TABLE

Event	***Trigger***	***Activity***	***Source***	***Response***	***Destination***
Customer enquires their problems	Enquiry about Reservation, Concession or Bus Details	Look up Concession or Reservation availability	Customer / Passenger	Status of problem and their solution	Customer/ Passenger
Passenger takes Reservation	Reservation details, Passenger details	Create Passenger Reservation details	Customer / Passenger	Passenger gets ticket number	Customer/ Passenger
Passenger Cancels or changes the Reservation	Cancellation	Update Cancellation record/ Create cancellation details	Customer / Passenger	Status of Cancellation	Customer/ Passenger
Student or Non-Student ask for concession	Concession enquiry	Create Concession details	Student/ Non-Student	Get concession Slip	Student/ Non-Student
Non-Student apply for Pass(Concession)	Non-Student Details	Creates pass details	Non-Student	Pass	Non-Student
Girls of School apply for pass	Girls details	Generates pass	School girls	Free Pass	School Girls

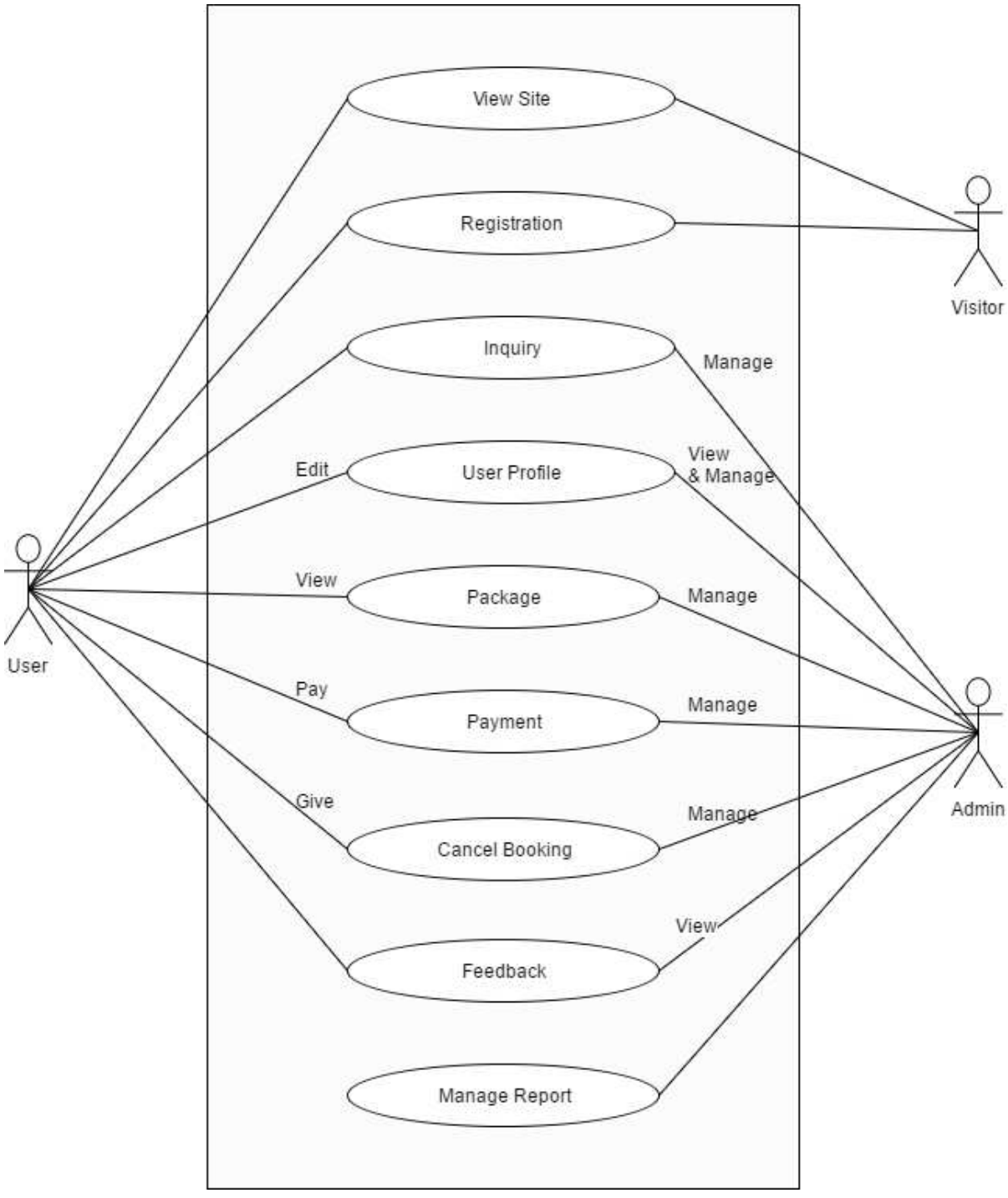
UML

DIAGRAMS

6.4.1 CLASS DIAGRAM



6.4.2 USE CASE DIAGRAM



×

6.4.4 STATE CHART

6.4.5 ACTIVITY DIAGRAM

6.4.6 MENU TREE

- Login
- MDI Form
- Employee Details
  - Driver's Info
  - Conductor's Info
- Bus Details
  - Long Route
  - Short Route
- Concession Details
  - Select
- Normal Concession
- Free Concession

- Reservation
  - Select
- Reservation
- Cancellation
- Pantry
- Report
  - Reservation
  - Cancellation
  - Concession
    - Utility
      - Calculator
      - Notepad
- Exit

6.4.7 DATABASE DESIGN

Login

Table - dbo.Login

Column Name	Data Type	Allow Nulls
LoginId	varchar(20)	
Password	varchar(20)	

Employee Master

Table - dbo.EmployeeMaster

Column Name	Data Type	Allow Nulls
EmpId	varchar(20)	
EmpName	varchar(30)	
EmpAdd	varchar(100)	
EmpDesignation	varchar(50)	
EmpPhno	numeric(18, 0)	
EmpMobno	numeric(18, 0)	

Diver Info

Table - dbo.DriverInfo

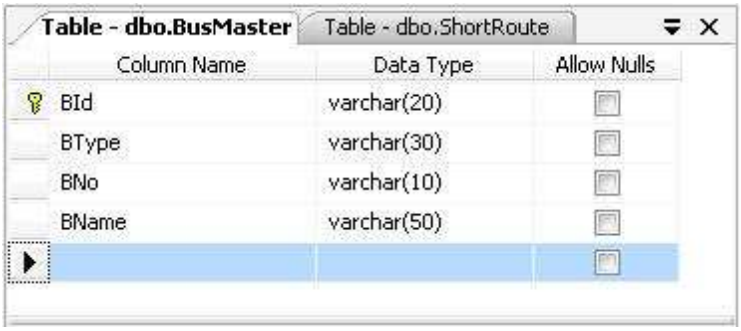
Column Name	Data Type	Allow Nulls
DId	varchar(20)	
DName	varchar(30)	
DAdd	varchar(100)	
DLisen	varchar(50)	
DPhno	numeric(18, 0)	
DMobno	numeric(18, 0)	

Conductor Info

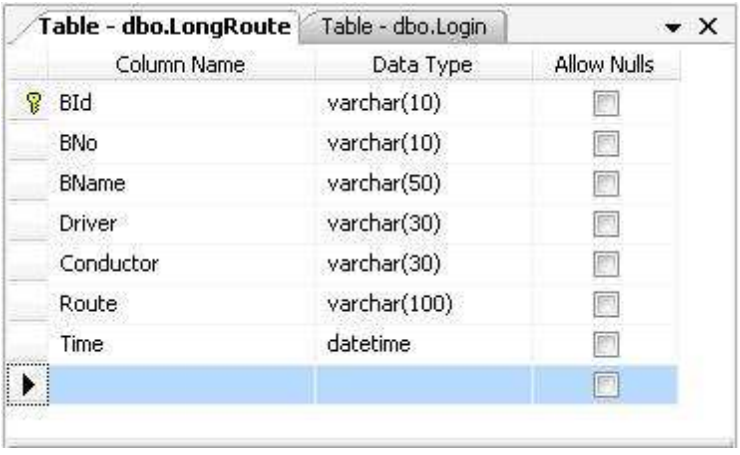
Table - dbo.ConductorInfo

Column Name	Data Type	Allow Nulls
CId	varchar(20)	
CName	varchar(30)	
CAdd	varchar(100)	
CPhno	numeric(18, 0)	
CMobno	numeric(18, 0)	

Bus Master



Long Route



Short Route

Table - dbo.ShortRoute | Table - dbo.Reservation

Column Name	Data Type	Allow Nulls
BId	varchar(20)	
BNo	varchar(10)	
BName	varchar(30)	
Driver	varchar(30)	
Conductor	varchar(30)	
Time	datetime	

CONCESSION

Normal Concession

Table - dbo.Concession | Table - dbo.Cancellation

Column Name	Data Type	Allow Nulls
ID	varchar(20)	
Name	varchar(30)	
Address	varchar(100)	
Dob	datetime	
Age	varchar(5)	
gen	varchar(6)	
concession	varchar(8)	
Destination	varchar(30)	
ValidFrom	datetime	
ValidTo	datetime	
PassRates	money	
Amount	money	

Free Concession

Table - dbo.FreeConcession

Column Name	Data Type	Allow Nulls
ID	varchar(20)	
Name	varchar(30)	
StuAdd	varchar(100)	
Dob	datetime	
Age	varchar(5)	
EduName	varchar(30)	
Destination	varchar(30)	
ValidFrom	datetime	
ValidTo	datetime	

Reservation

Table - dbo.Reservation | Table - dbo.Pantry

Column Name	Data Type	Allow Nulls
ResId	varchar(20)	
Name	varchar(30)	
Address	varchar(100)	
BNo	varchar(10)	
gen	varchar(6)	
Type	varchar(14)	
Destination	varchar(30)	
SeatNo	varchar(20)	
Time	datetime	
Date	datetime	
TicketRates	money	
Amount	money	

Cancellation

Table - dbo.Cancellation | Table - dbo.BusMaster

Column Name	Data Type	Allow Nulls
ResID	varchar(20)	
Name	varchar(30)	
Address	varchar(100)	
BNo	varchar(10)	
gen	varchar(6)	
Type	varchar(14)	
Destination	varchar(50)	
Date	datetime	
Time	datetime	
Today	datetime	
TicketRates	money	
Amount	money	
Refund	money	

6.4.8 CRUD TABLE

Table Form	Employee_Details	Driver_Info	Conductor_Info	Bus_Master	Long_Route	Short_Route	Normal_Concession	Free_Concession	Reservation	Cancellation
Employee details	CRUD									
Driver Info		CRUD								
Conductor Info			CRUD							
Bus Master				CRUD						
Long Route					CRUD					
Short Route						CRUD				
Normal Concession							CRUD			
Free Concession								CRUD		

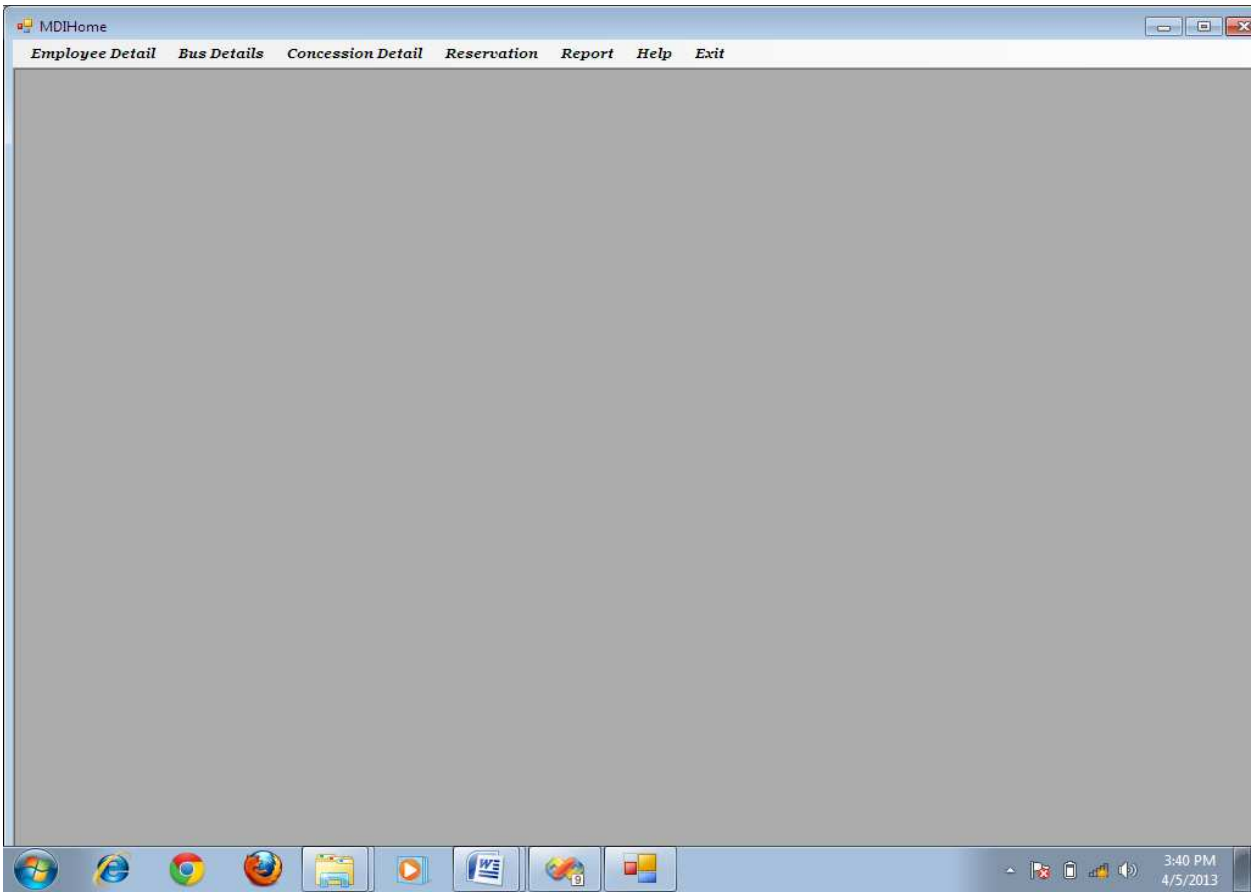
Reservation									CR UD	
Cancellation										CRUD

SCREENS LAYOUT

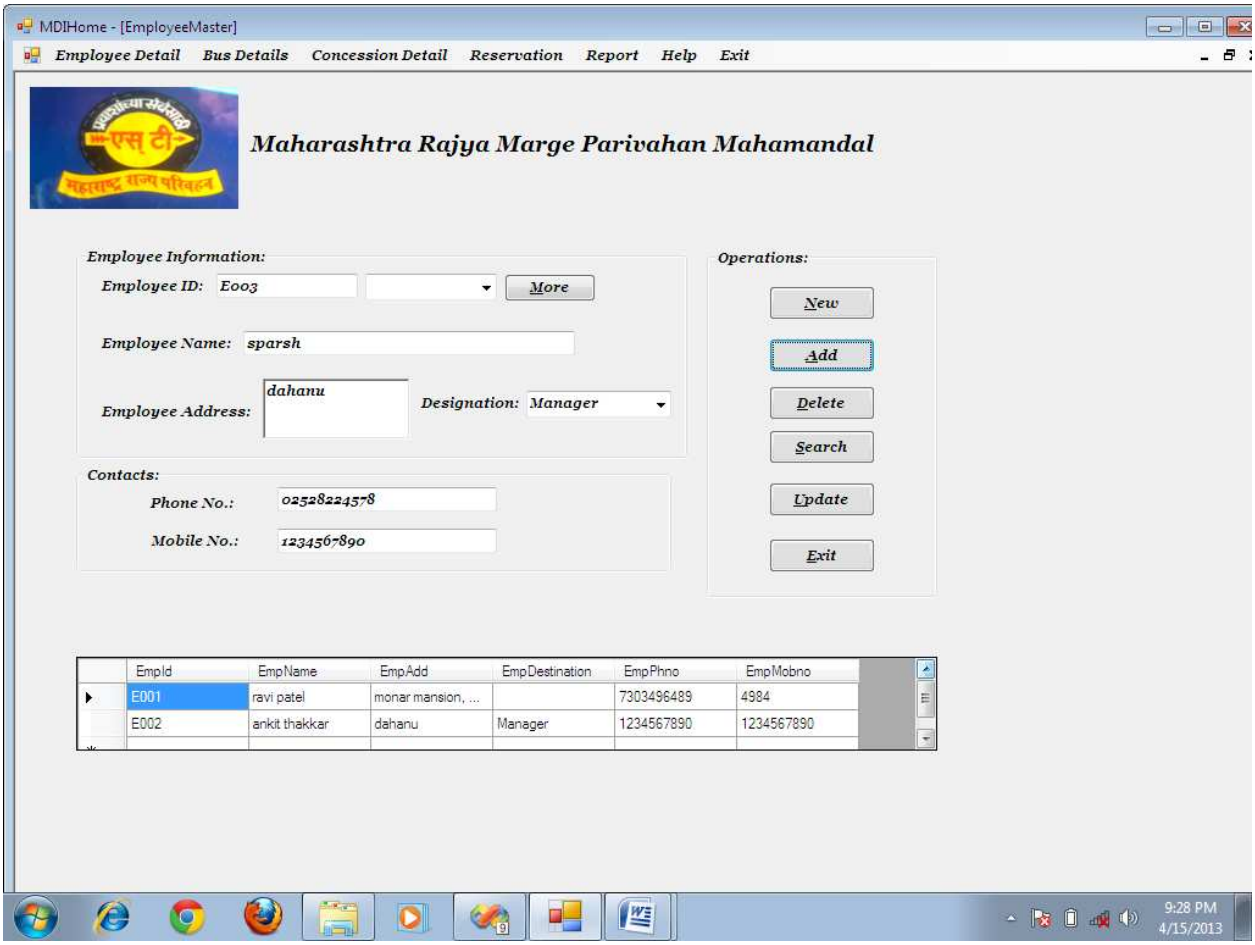
LOGIN SCREEN

WELCOME SCREEN

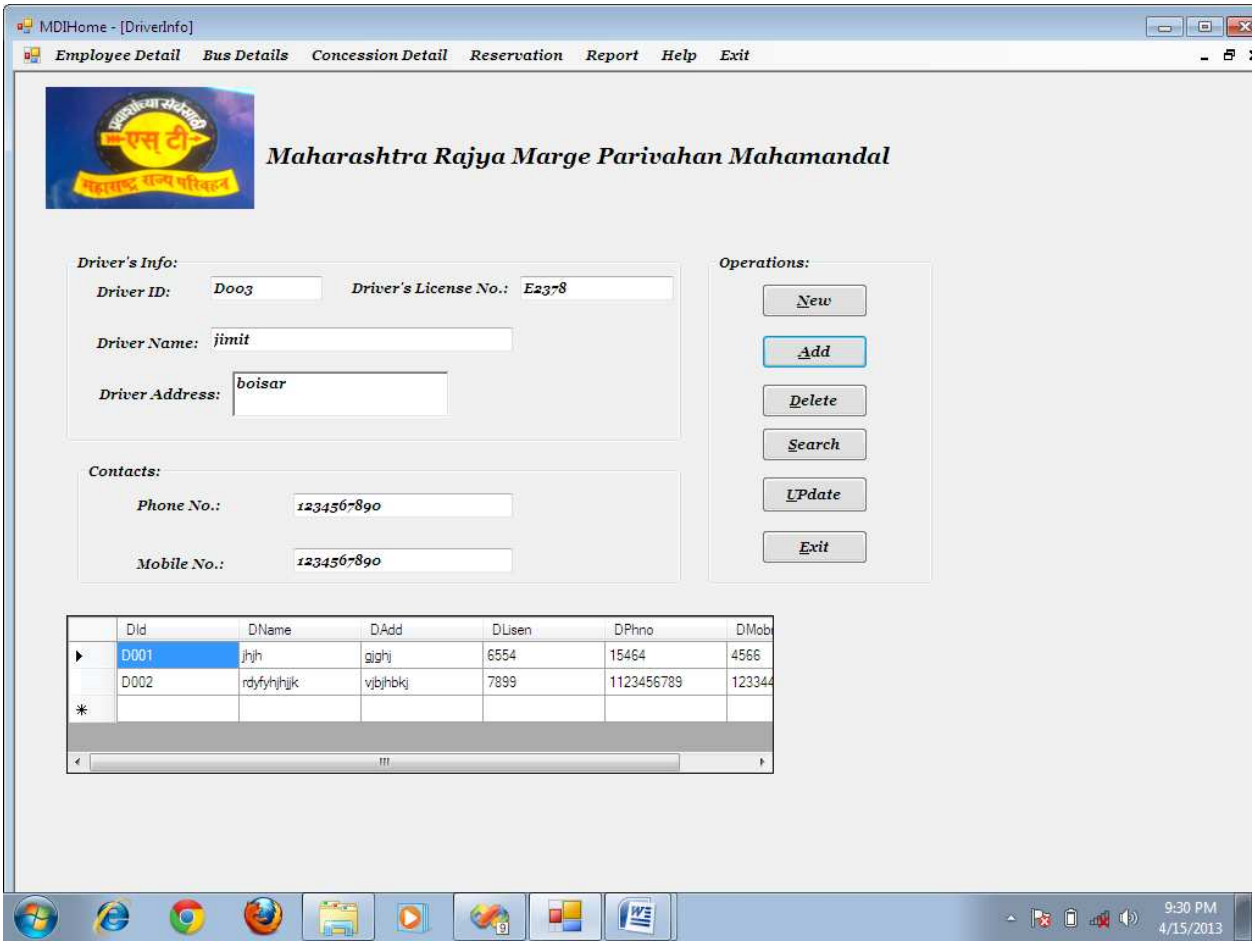
MDI FORM



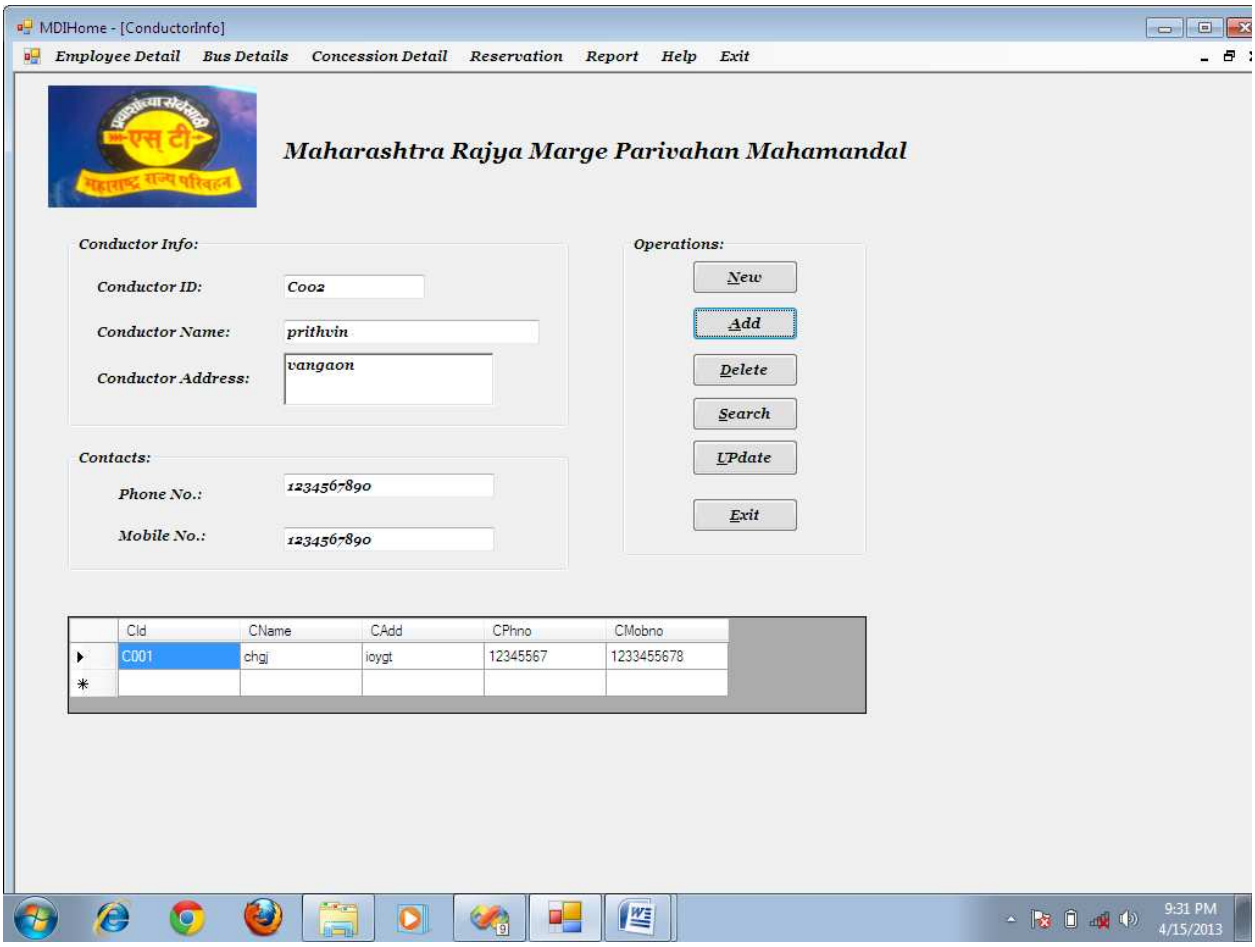
EMPLOYEE MASTER



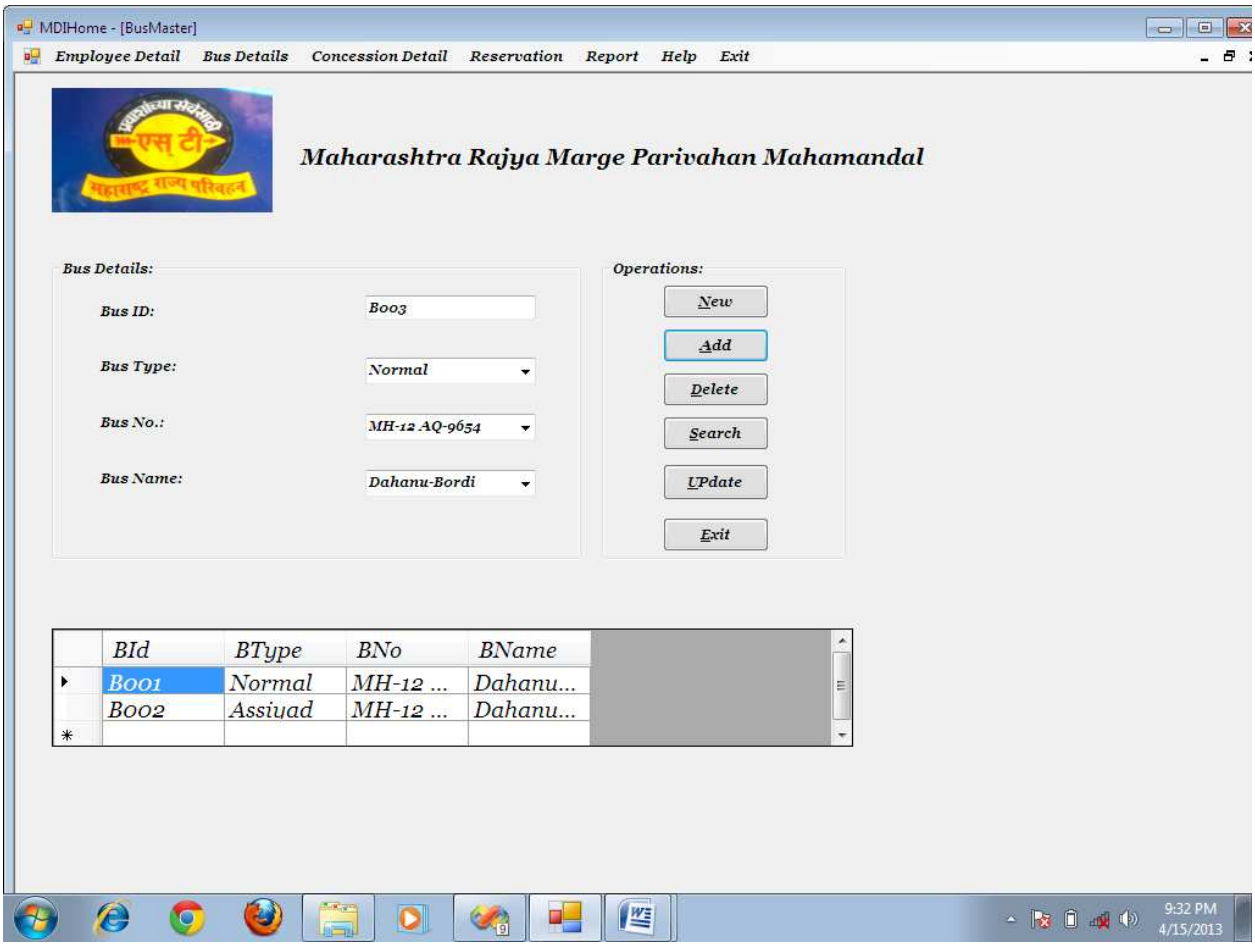
DRIVER'S INFORMATION



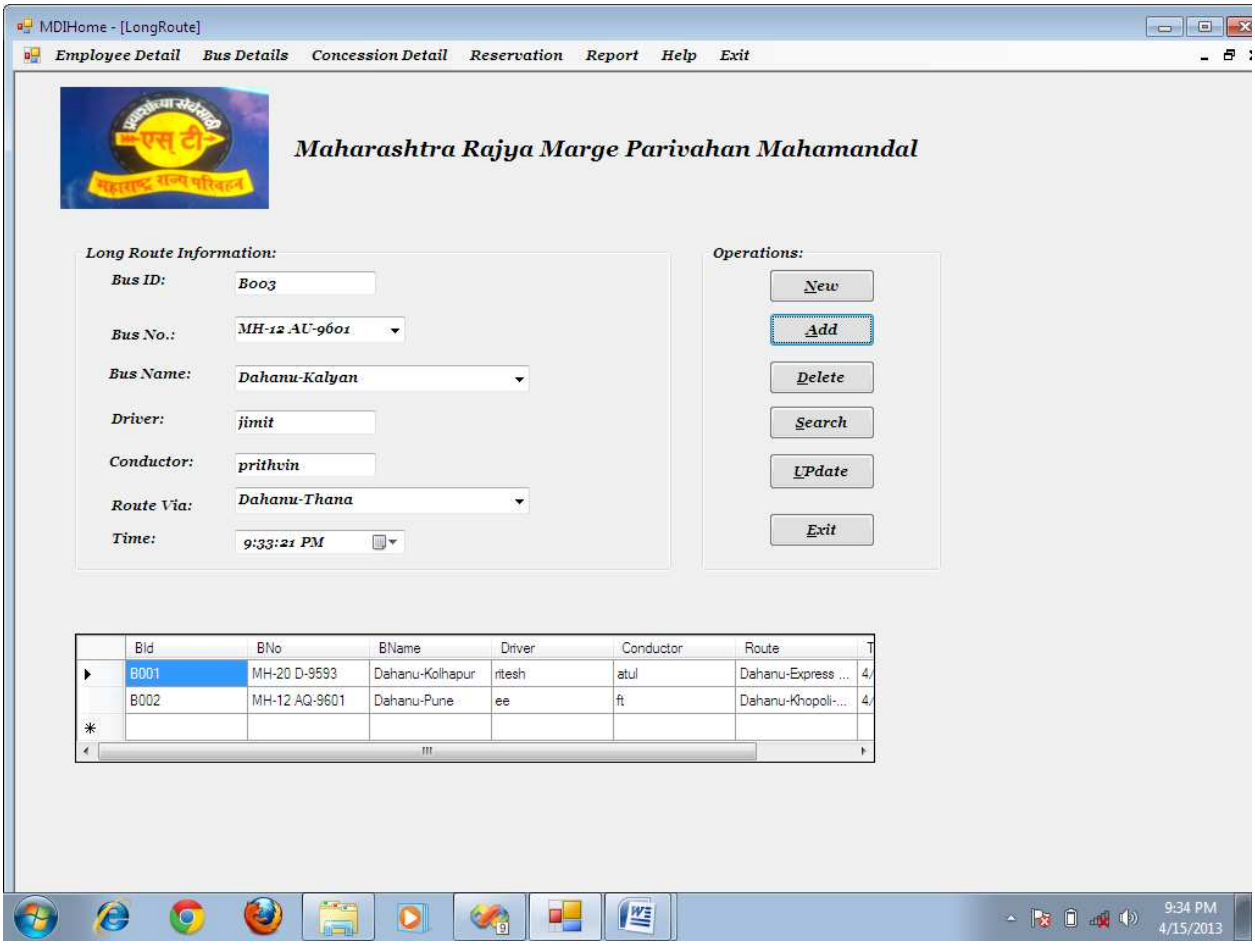
CONDUCTOR'S INFORMATION


MDIHome - [ConductorInfo]
Employee Detail
Bus Details
Concession Detail
Reservation
Report
Help
Exit
Maharashtra Rajya Marge Parivahan Mahamandal
Conductor Info:
Conductor ID:
C002
Conductor Name:
prithvin
Conductor Address:
vangaon
Contacts:
Phone No.:
1234567890
Mobile No.:
1234567890
Operations:
New
Add
Delete
Search
UPdate
Exit
CId
CName
CAdd
CPhno
CMobno
C001
chgj
ioygt
12345567
1233455678
9:31 PM
4/15/2013

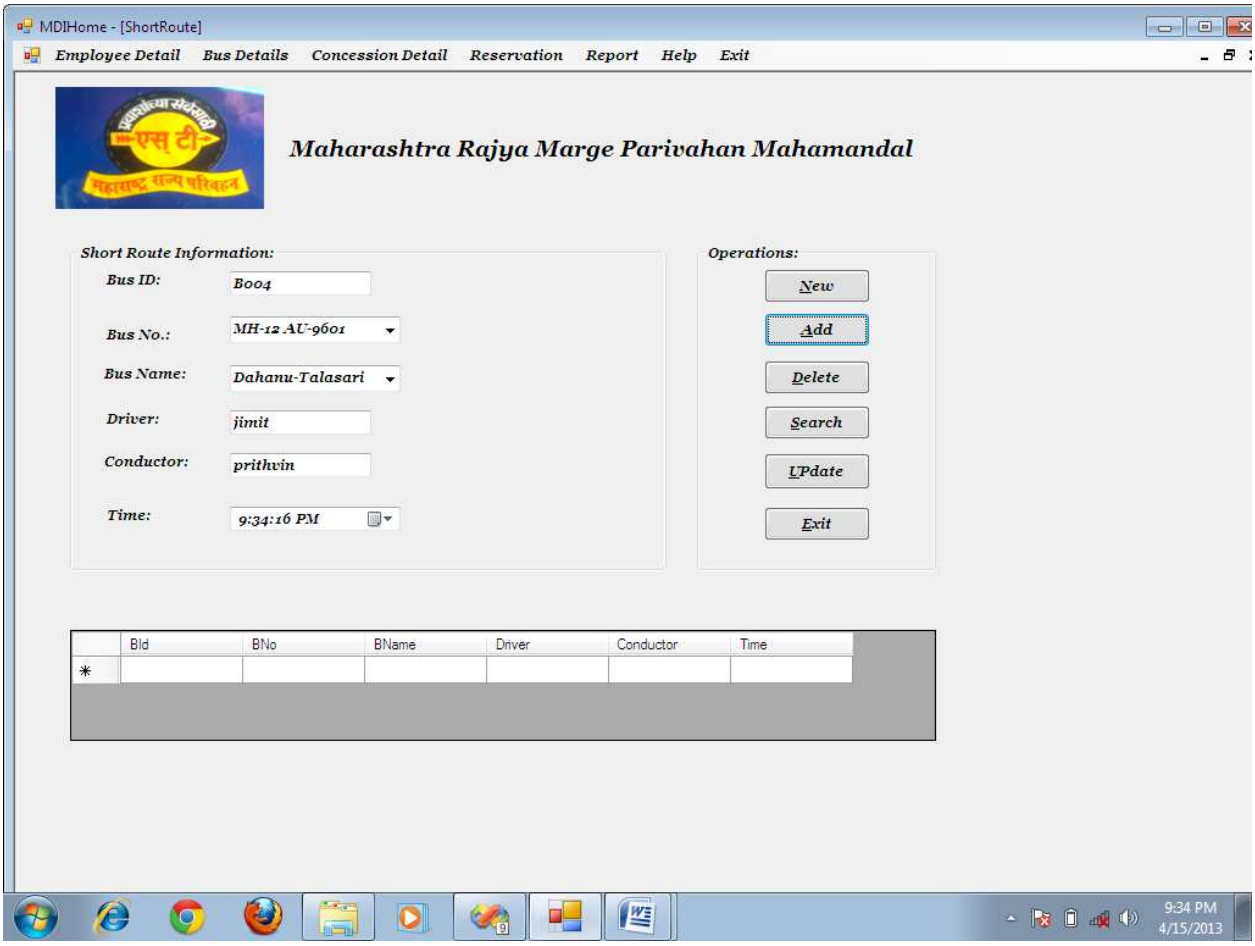
BUS MASTER


MDIHome - [BusMaster]
Employee Detail
Bus Details
Concession Detail
Reservation
Report
Help
Exit
Maharashtra Rajya Marge Parivahan Mahamandal
Bus Details:
Bus ID:
B003
Bus Type:
Normal
Bus No.:
MH-12 AQ-9654
Bus Name:
Dahanu-Bordi
Operations:
New
Add
Delete
Search
UPdate
Exit
BId
BType
BNo
BName
B001
Normal
MH-12 ...
Dahanu...
B002
Assiyad
MH-12 ...
Dahanu...
9:32 PM
4/15/2013

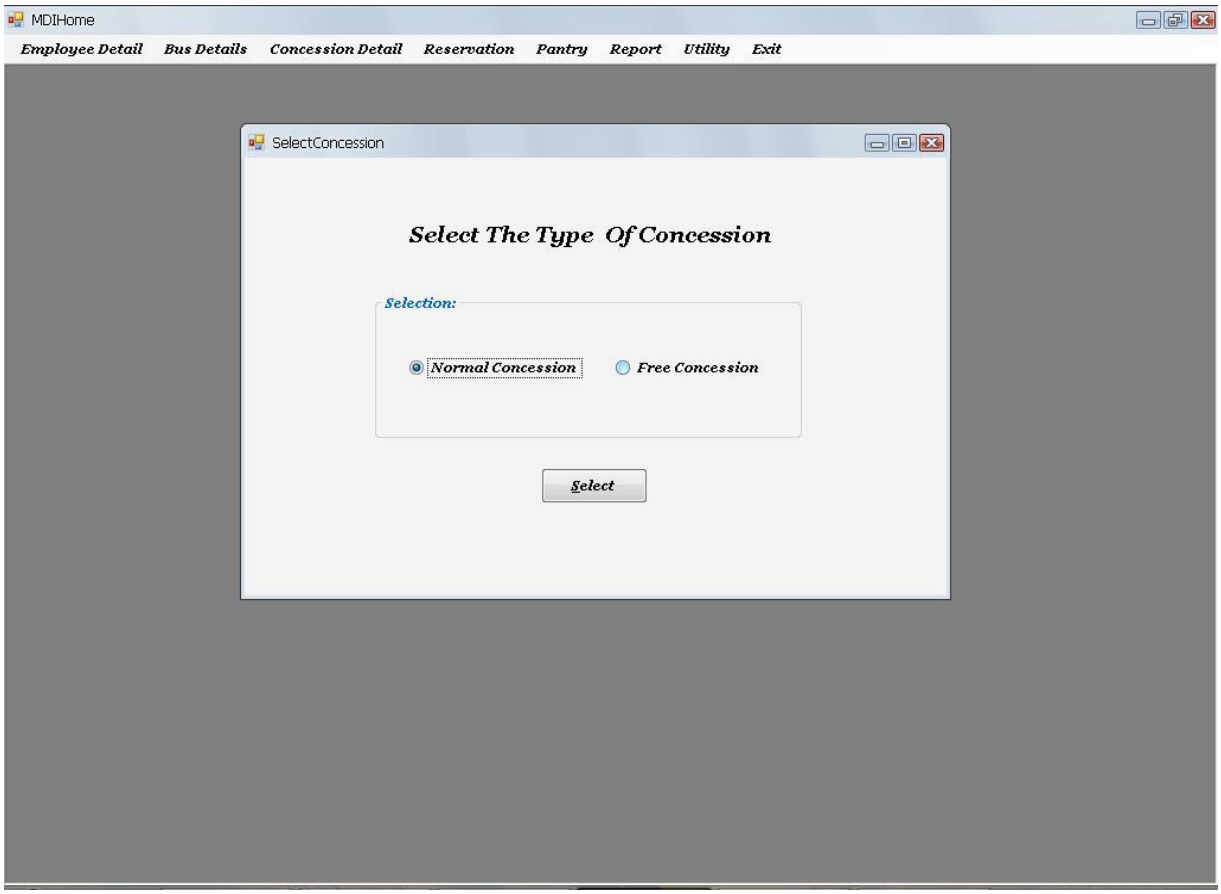
LONG ROUTE INFORMATION



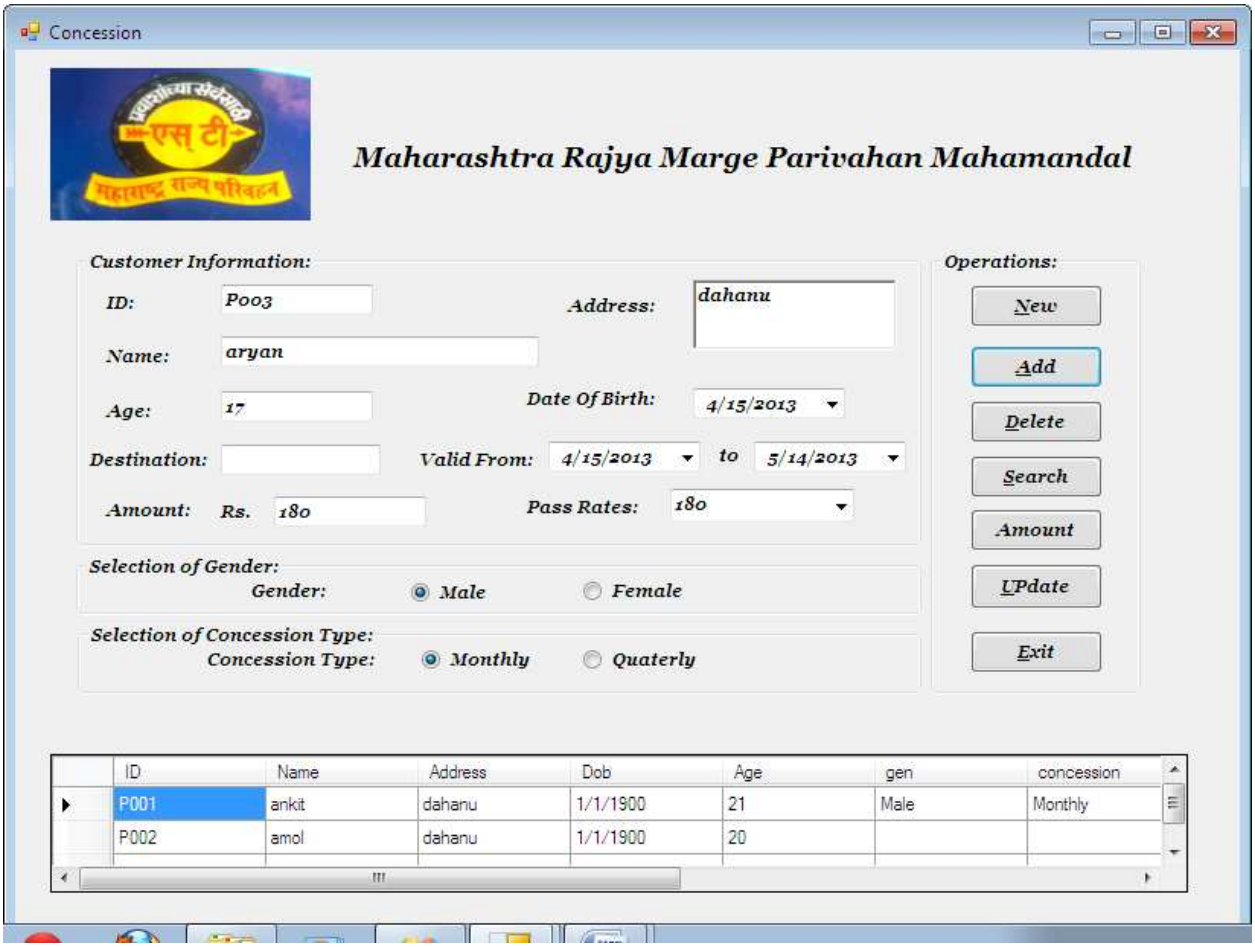
SHORT ROUTE INFORMATION



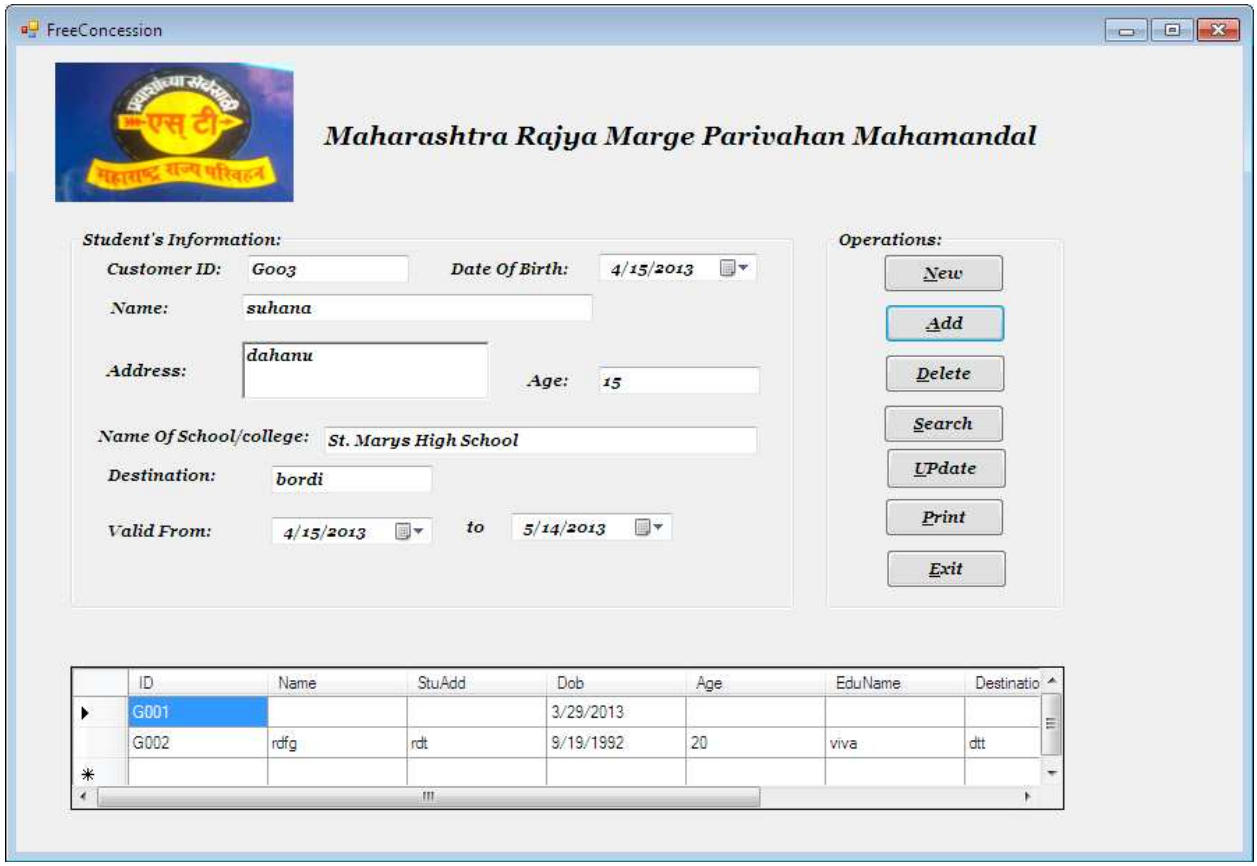
SELECTION OF CONCESSION



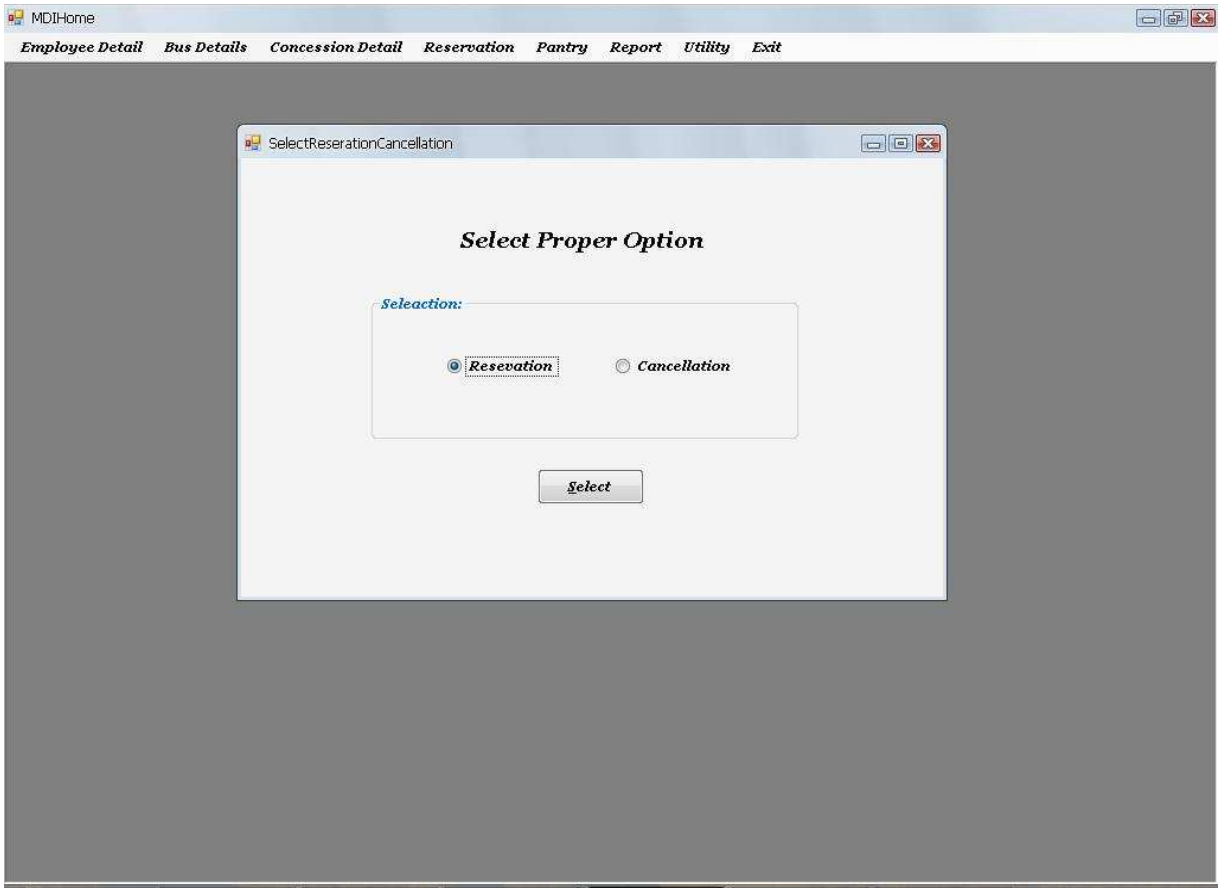
NORMAL CONCESSION PASS



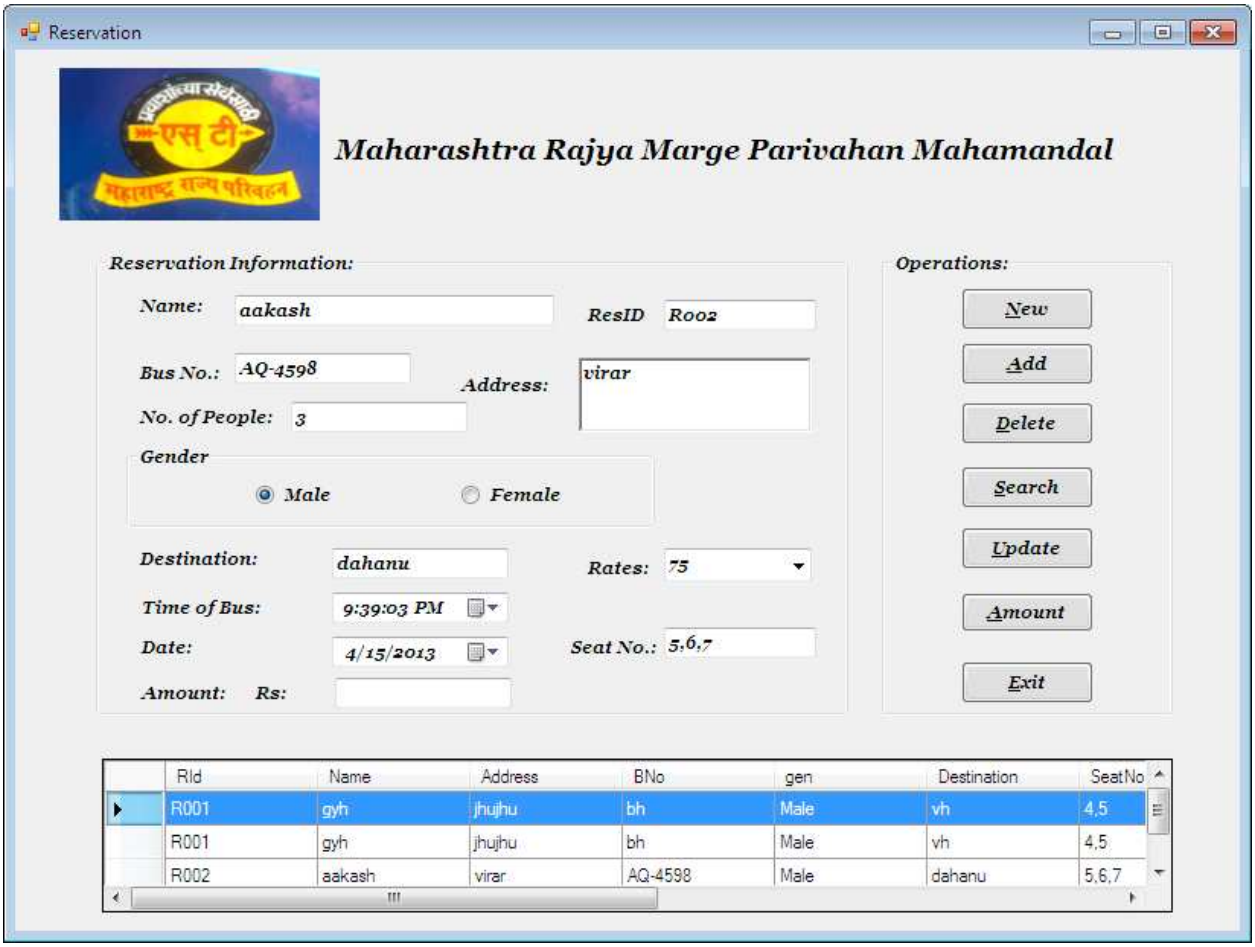
FREE CONCESSION PASS



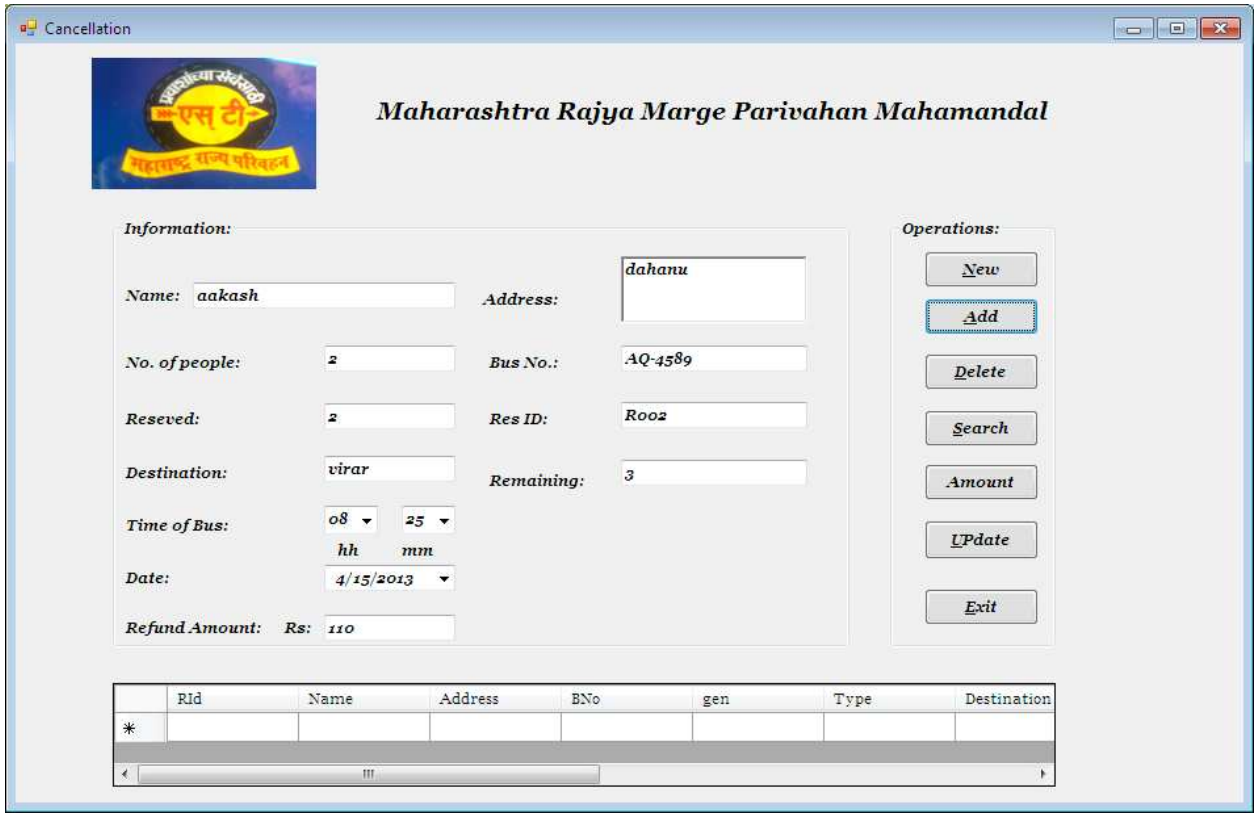
SELECTION FOR RESERVATION/CANCELLATION



RESERVATION FORM



CANCELLATION



CODE

Imports System.Data.SqlClient

```
Imports System.Collections.Generic

Public Class DriverInfo
    Dim cn As New SqlConnection("Data Source=.;Initial
Catalog=BusSystemManagement;Integrated Security=True")
    Dim cmd As SqlCommand
    Dim dr As SqlDataReader
    Dim ds As New DataSet
    Dim dv As New DataView
    Dim da As New SqlDataAdapter
    Dim id As String = "D00"
    Dim num As Integer = 0

    Private Sub DriverInfo_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        txtDriverID.Text = ""
        txtDriverName.Text = ""
        txtLicenseNo.Text = ""
        txtDriverAddress.Text = ""
        txtPhoneNo.Text = ""
        txtMobileNo.Text = ""
        txtDriverID.Enabled = True
        txtDriverName.Enabled = False
        txtLicenseNo.Enabled = False
        txtDriverAddress.Enabled = False
        txtPhoneNo.Enabled = False
        txtMobileNo.Enabled = False
        btnAdd.Enabled = False
        btnDelete.Enabled = False
```

```
        btnUpdate.Enabled = False
        btnBack.Enabled = False
        Me.txtDriverID.Focus()

    End Sub
    Public Function getId(ByVal num As Integer)
        id = "D00"
        If num <= 9 Then
            id = id & num
        Else
            If num <= 99 Then
                id = id.Substring(0, 2)
                id = id & num

            End If

        End If
        Return id
    End Function

    Private Sub btnNew_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnNew.Click
        txtDriverName.Text = ""
        txtLicenseNo.Text = ""
        txtDriverAddress.Text = ""
        txtPhoneNo.Text = ""
        txtMobileNo.Text = ""
        txtDriverID.Enabled = False
        txtDriverName.Enabled = True
        txtLicenseNo.Enabled = True
```

```
        txtDriverAddress.Enabled = True
        txtPhoneNo.Enabled = True
        txtMobileNo.Enabled = True
        btnNew.Enabled = False
        btnAdd.Enabled = True
        btnBack.Enabled = True
        btnDelete.Enabled = False
        btnSearch.Enabled = False
        btnUpdate.Enabled = False
        Try
            cn.Close()
            cn.Open()
            Dim query As String
            query = "select * from DriverInfo"
            cmd = New SqlCommand(query, cn)
            da.SelectCommand = cmd
            da.Fill(ds, "DriverInfo")
            dv = New DataView(ds.Tables("DriverInfo"))
            Me.DataGridView1.DataSource = dv
            cn.Close()
        Catch ex As Exception
        End Try

        Dim str, query1 As String
        query1 = "select max(cast(substring(DId,3,4)as int))from
DriverInfo"
        cmd = New SqlCommand(query1, cn)
        cn.Open()
        str = cmd.ExecuteScalar.ToString()
        If str = "" Then
```

```
            num = num + 1
        Else
            num = str
            num = str + 1
        End If
        getId(num)
        txtDriverID.Text = id
        cn.Close()

    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAdd.Click
        If (txtDriverID.Text <> "" And txtDriverName.Text <> ""
And txtLicenseNo.Text <> "" And txtDriverAddress.Text <> ""
And txtPhoneNo.Text <> "" And txtMobileNo.Text <> "")
Then
            Try
                cn.Close()
                cn.Open()
                Dim query As String
                query = "insert into DriverInfo values('" &
Me.txtDriverID.Text & "','" & Me.txtDriverName.Text & "','"
& Me.txtDriverAddress.Text & "','" & Me.txtLicenseNo.Text
& "','" & Me.txtPhoneNo.Text & "','" & Me.txtMobileNo.Text
& "')"
                cmd = New SqlCommand(query, cn)
                cmd.ExecuteNonQuery()
                MessageBox.Show("Details are added.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
            txtDriverID.Text = ""
            txtDriverName.Text = ""
            txtLicenseNo.Text = ""
            txtDriverAddress.Text = ""
            txtPhoneNo.Text = ""
            txtMobileNo.Text = ""
            txtDriverID.Enabled = True
            txtDriverName.Enabled = False
            txtLicenseNo.Enabled = False
            txtDriverAddress.Enabled = False
            txtPhoneNo.Enabled = False
            txtMobileNo.Enabled = False
            btnNew.Enabled = True
            btnAdd.Enabled = False
            btnDelete.Enabled = False
            btnUpdate.Enabled = False
            btnSearch.Enabled = True
            btnBack.Enabled = False
            Me.txtDriverID.Focus()
        Else
            MsgBox("PLEASE ENSURE THAT ALL FIELDS
ARE COMPLETE....!", MsgBoxStyle.Information +
MsgBoxStyle.OkOnly)
        End If
        cn.Close()
```

```
    End Sub

    Private Sub btnDelete_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
        Try
            cn.Close()
            cn.Open()
            Dim query As String
            query = "delete from DriverInfo where DId='" &
Me.txtDriverID.Text & "'"
            cmd = New SqlCommand(query, cn)
            cmd.ExecuteNonQuery()
            MessageBox.Show("Details are deleted.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        cn.Close()

        txtDriverID.Text = ""
        txtDriverName.Text = ""
        txtLicenseNo.Text = ""
        txtDriverAddress.Text = ""
        txtPhoneNo.Text = ""
        txtMobileNo.Text = ""
        txtDriverID.Enabled = True
        txtDriverName.Enabled = False
        txtLicenseNo.Enabled = False
```

```
        txtDriverAddress.Enabled = False
        txtPhoneNo.Enabled = False
        txtMobileNo.Enabled = False
        btnNew.Enabled = True
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnUpdate.Enabled = False
        btnSearch.Enabled = True
        btnBack.Enabled = False
        Me.txtDriverID.Focus()
    End Sub

    Private Sub btnSearch_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnSearch.Click
        cn.Close()
        cn.Open()
        Dim query As String
        Dim cmd1 As SqlCommand
        query = "select * from DriverInfo where DId='" &
Me.txtDriverID.Text & "'"
        cmd1 = New SqlCommand(query, cn)
        dr = cmd1.ExecuteReader()
        If dr.HasRows = True Then
            dr.Read()
            Me.txtDriverName.Text = dr("DName")
            Me.txtDriverAddress.Text = dr("DAdd")
            Me.txtLicenseNo.Text = dr("DLisen")
            Me.txtPhoneNo.Text = dr("DPhno")
            Me.txtMobileNo.Text = dr("DMobno")
```

```
        Else
            MessageBox.Show("Details not found.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
        cn.Close()

        txtDriverID.Enabled = False
        txtDriverName.Enabled = True
        txtLicenseNo.Enabled = True
        txtDriverAddress.Enabled = True
        txtPhoneNo.Enabled = True
        txtMobileNo.Enabled = True
        btnNew.Enabled = False
        btnAdd.Enabled = False
        btnBack.Enabled = True
        btnDelete.Enabled = True
        btnUpdate.Enabled = True
        btnSearch.Enabled = False

    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnExit.Click
        Me.Close()

    End Sub

    Private Sub txtDriverID_KeyUp(ByVal sender As Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
txtDriverID.KeyUp
```

```
        If txtDriverID.Text Like
"[`~@#$%^&*()_+.:'"">><,/?|\[=!-]" Or txtDriverID.Text Like
"*[`~@#$%^&*()_+.:'"">><,/?|\[=!-]*" Then
            MessageBox.Show("Please Enter Number Only",
"Error")
            txtDriverID.Clear()
        End If

    End Sub

    Private Sub txtDriverName_KeyUp(ByVal sender As
Object, ByVal e As System.Windows.Forms.KeyEventArgs)
Handles txtDriverName.KeyUp
        If txtDriverName.Text Like "[0-9]" Or
txtDriverName.Text Like "*[0-9]*" Or txtDriverName.Text
Like "[`~@#$%^&*()_+.:,><}{;'/?|\}{[!=-]" Or
txtDriverName.Text Like "*[`~@#$%^&*()_+.:,><}{;'/?|\[!=-
]*" Then
            MessageBox.Show("Please Enter Alphabets Only",
"Error")
            txtDriverName.Clear()
        End If

    End Sub

    Private Sub txtDriverAddress_KeyUp(ByVal sender As
Object, ByVal e As System.Windows.Forms.KeyEventArgs)
Handles txtDriverAddress.KeyUp
```

```
        If txtDriverAddress.Text Like
"[`~@#$%^&*_+:><?|/""}{[!=]" Or txtDriverAddress.Text
Like "*[`~@#$%^&*_+:><}{?|/[!=]*" Then
            MessageBox.Show("Please Enter Alphabets &
Numbers Only", "Error")
            txtDriverAddress.Clear()
        End If

    End Sub

    Private Sub txtPhoneNo_KeyUp(ByVal sender As Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
txtPhoneNo.KeyUp
        If (IsNumeric(txtPhoneNo.Text) = False) Then
            MessageBox.Show("Please Enter Number Only",
"Error")
            txtPhoneNo.Clear()
        End If

    End Sub

    Private Sub txtMobileNo_KeyUp(ByVal sender As Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
txtMobileNo.KeyUp
        If (IsNumeric(txtMobileNo.Text) = False) Then
            MessageBox.Show("Please Enter Number Only",
"Error")
            txtMobileNo.Clear()
        End If
```

```
    End Sub

    Private Sub btnUpdate_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnUpdate.Click
        Try
            cn.Close()
            cn.Open()
            Dim str As String
            str = "update DriverInfo set  DName='" &
txtDriverName.Text & "',DAdd='" & txtDriverAddress.Text &
"',DLisen='" & txtDriverLicenseNo.Text & "',DPhNo='" &
txtPhoneNo.Text & "',DMobNo='" & txtMobileNo.Text & "'
where DId='" & txtDriverID.Text & "'"
            cmd = New SqlCommand(str, cn)

            cmd.ExecuteNonQuery()
            MessageBox.Show("Details are updated", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try

        txtDriverID.Text = ""
        txtDriverName.Text = ""
        txtLicenseNo.Text = ""
        txtDriverAddress.Text = ""
        txtPhoneNo.Text = ""
        txtMobileNo.Text = ""
```

```
        txtDriverID.Enabled = True
        txtDriverName.Enabled = False
        txtLicenseNo.Enabled = False
        txtDriverAddress.Enabled = False
        txtPhoneNo.Enabled = False
        txtMobileNo.Enabled = False
        btnNew.Enabled = True
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnUpdate.Enabled = False
        btnSearch.Enabled = True
        btnBack.Enabled = False
        Me.txtDriverID.Focus()

    End Sub

    Private Sub DataGridView1_CellClick1(ByVal sender As
Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridView1.CellClick
        Me.txtDriverID.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("DId").Value
        Me.txtDriverAddress.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("DAdd").Value
        Me.txtLicenseNo.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("DLisen").Value
```

```
        Me.txtMobileNo.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("DMobno").Value
        Me.txtPhoneNo.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("DPhno").Value
        Me.txtDriverName.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("DName").Value
    End Sub

    Private Sub txtLicenseNo_KeyUp(ByVal sender As Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
txtLicenseNo.KeyUp
        If txtDriverID.Text Like
"[`~@#$%^&*()_+.:'""><,/?|\[=!-]" Or txtDriverID.Text Like
"*[`~@#$%^&*()_+.:'""><,/?|\[=!-]*" Then
            MessageBox.Show("Please Enter Number &
Characters Only", "Error")
            txtDriverID.Clear()
        End If
    End Sub

    Private Sub btnBack_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnBack.Click
        txtDriverID.Text = ""
        txtDriverName.Text = ""
        txtLicenseNo.Text = ""
        txtDriverAddress.Text = ""
        txtPhoneNo.Text = ""
```

```
        txtMobileNo.Text = ""
        txtDriverID.Enabled = True
        txtDriverName.Enabled = False
        txtLicenseNo.Enabled = False
        txtDriverAddress.Enabled = False
        txtPhoneNo.Enabled = False
        txtMobileNo.Enabled = False
        btnNew.Enabled = True
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnUpdate.Enabled = False
        btnSearch.Enabled = True
        btnBack.Enabled = False
    End Sub
End Class
```

CODE

```
Imports System.Data.SqlClient
Imports System.Collections.Generic
Public Class BusMaster
    Dim cn As New SqlConnection("Data Source=.;Initial
Catalog=BusSystemManagement;Integrated Security=True")
    Dim cmd As SqlCommand
    Dim dr As SqlDataReader
    Dim ds As New DataSet
    Dim dv As New DataView
    Dim da As New SqlDataAdapter
    Dim id As String = "B00"
    Dim num As Integer = 0
    Public Function getId(ByVal num As Integer)
```

```
        id = "B00"
        If num <= 9 Then
            id = id & num
        Else
            If num <= 99 Then
                id = id.Substring(0, 2)
                id = id & num
            End If
        End If
        Return id
    End Function

    Private Sub btnNew_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnNew.Click
        cboBusType.Text = ""
        cboBusNo.Text = ""
        cboBusName.Text = ""
        cboBusType.Enabled = True
        cboBusNo.Enabled = True
        cboBusName.Enabled = True
        txtBusID.Enabled = False
        btnNew.Enabled = False
        btnAdd.Enabled = True
        btnBack.Enabled = True
        btnDelete.Enabled = False
        btnSearch.Enabled = False
        btnUpdate.Enabled = False

        Try
            cn.Close()
```

```
            cn.Open()
            Dim query As String
            query = "select * from BusMaster"
            cmd = New SqlCommand(query, cn)
            da.SelectCommand = cmd
            da.Fill(ds, "BusMaster")
            dv = New DataView(ds.Tables("BusMaster"))
            Me.DataGridView1.DataSource = dv
            cn.Close()
        Catch ex As Exception
        End Try

        Dim str, query1 As String
        query1 = "select max(cast(substring(BId,3,4)as int))from
BusMaster"
        cmd = New SqlCommand(query1, cn)
        cn.Open()
        str = cmd.ExecuteScalar.ToString()
        If str = "" Then
            num = num + 1
        Else
            num = str
            num = str + 1
        End If
        getId(num)
        txtBusID.Text = id
        cn.Close()

    End Sub
```

```
    Private Sub btnAdd_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAdd.Click
        If (txtBusID.Text <> "" And cboBusName.Text <> "" And
cboBusNo.Text <> "" And cboBusType.Text <> "") Then
            Try
                cn.Close()
                cn.Open()
                Dim query As String
                query = "insert into BusMaster values('" &
Me.txtBusID.Text & "','" & Me.cboBusType.SelectedItem &
"','" & Me.cboBusNo.SelectedItem & "','" &
Me.cboBusName.SelectedItem & "')"
                cmd = New SqlCommand(query, cn)
                cmd.ExecuteNonQuery()
                MessageBox.Show("Details are added.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
                MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try

            txtBusID.Text = ""
            cboBusType.Text = ""
            cboBusNo.Text = ""
            cboBusName.Text = ""
            txtBusID.Enabled = True
            cboBusType.Enabled = False
            cboBusNo.Enabled = False
            cboBusName.Enabled = False
            btnAdd.Enabled = False
```

```
            btnDelete.Enabled = False
            btnUpdate.Enabled = False
            btnBack.Enabled = False
            btnNew.Enabled = True
            btnSearch.Enabled = True
            Me.txtBusID.Focus()
        Else
            MsgBox("PLEASE ENSURE THAT ALL FIELDS
ARE COMPLETE....!", MsgBoxStyle.Information +
MsgBoxStyle.OkOnly)

        End If
        cn.Close()
    End Sub

    Private Sub btnDelete_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
        Try
            cn.Close()
            cn.Open()
            Dim query As String
            query = "delete from BusMaster where BId='" &
Me.txtBusID.Text & "'"
            cmd = New SqlCommand(query, cn)
            cmd.ExecuteNonQuery()
            MessageBox.Show("Details are deleted.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
```

```
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        cn.Close()
        txtBusID.Text = ""
        cboBusType.Text = ""
        cboBusNo.Text = ""
        cboBusName.Text = ""
        txtBusID.Enabled = True
        cboBusType.Enabled = False
        cboBusNo.Enabled = False
        cboBusName.Enabled = False
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnUpdate.Enabled = False
        btnBack.Enabled = False
        btnNew.Enabled = True
        btnSearch.Enabled = True
        Me.txtBusID.Focus()

    End Sub
    Private Sub btnSearch_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnSearch.Click
        cn.Close()
        cn.Open()
        Dim query As String
        query = "select * from BusMaster where BId='" &
Me.txtBusID.Text & "'"
        cmd = New SqlCommand(query, cn)
```

```
        dr = cmd.ExecuteReader()
        If dr.HasRows = True Then
            dr.Read()
            Me.cboBusType.Text = dr("BType")
            Me.cboBusNo.Text = dr("BNo")
            Me.cboBusName.Text = dr("BName")
        Else
            MessageBox.Show("Details not found.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
        cn.Close()
        btnAdd.Enabled = False
        btnDelete.Enabled = True
        btnUpdate.Enabled = True
        btnBack.Enabled = True
        btnNew.Enabled = False
        btnSearch.Enabled = False
        txtBusID.Enabled = False
        cboBusType.Enabled = True
        cboBusNo.Enabled = True
        cboBusName.Enabled = True
    End Sub

    Private Sub btnExit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnExit.Click
        Me.Close()
    End Sub
```

```
    Private Sub btnUpdate_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnUpdate.Click
        Try
            cn.Close()
            cn.Open()
            Dim str As String
            str = "update BusMaster set BType='" &
cboBusType.Text & "', BNo='" & cboBusNo.Text &
"',BName='" & cboBusName.Text & "'where BId='" &
txtBusID.Text & "'"
            cmd = New SqlCommand(str, cn)
            cmd.ExecuteNonQuery()
            MessageBox.Show("Details are updated", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        cn.Close()
        txtBusID.Text = ""
        cboBusType.Text = ""
        cboBusNo.Text = ""
        cboBusName.Text = ""
        txtBusID.Enabled = True
        cboBusType.Enabled = False
        cboBusNo.Enabled = False
        cboBusName.Enabled = False
        btnAdd.Enabled = False
        btnDelete.Enabled = False
```

```
        btnUpdate.Enabled = False
        btnBack.Enabled = False
        btnNew.Enabled = True
        btnSearch.Enabled = True
        Me.txtBusID.Focus()
    End Sub

    Private Sub DataGridView1_CellClick(ByVal sender As
Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
        Me.txtBusID.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BId").Value
        Me.cboBusType.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BType").Value
        Me.cboBusNo.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BNo").Value
        Me.cboBusName.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BName").Value
        DataGridView1.Refresh()
    End Sub

    Private Sub DataGridView1_CellContentClick(ByVal sender
As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
    End Sub
```

```
    Private Sub DataGridView1_CellClick1(ByVal sender As
Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridView1.CellClick
        Me.txtBusID.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BId").Value
        Me.cboBusType.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BType").Value
        Me.cboBusNo.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BNo").Value
        Me.cboBusName.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BName").Value
        DataGridView1.Refresh()
    End Sub

    Private Sub BusMaster_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        txtBusID.Text = ""
        cboBusType.Text = ""
        cboBusNo.Text = ""
        cboBusName.Text = ""
        txtBusID.Enabled = True
        cboBusType.Enabled = False
        cboBusNo.Enabled = False
        cboBusName.Enabled = False
```

```
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnUpdate.Enabled = False
        btnBack.Enabled = False
        Me.txtBusID.Focus()
    End Sub

    Private Sub btnBack_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnBack.Click
        txtBusID.Text = ""
        cboBusType.Text = ""
        cboBusNo.Text = ""
        cboBusName.Text = ""
        txtBusID.Enabled = True
        cboBusType.Enabled = False
        cboBusNo.Enabled = False
        cboBusName.Enabled = False
        btnAdd.Enabled = False
        btnDelete.Enabled = False
        btnUpdate.Enabled = False
        btnBack.Enabled = False
        btnNew.Enabled = True
        btnSearch.Enabled = True

    End Sub
End Class
```

CODE

```
Imports System.Data.SqlClient
```

```
Imports System.Collections.Generic

Public Class Cancellation
    Dim cn As New SqlConnection("Data Source=.;Initial
Catalog=BusSystemManagement;Integrated Security=True")
    Dim cmd As SqlCommand
    Dim dr As SqlDataReader
    Dim ds As New DataSet
    Dim dv As New DataView
    Dim da As New SqlDataAdapter
    Dim dpre As New Date
    Dim price, money As Double
    Dim no, dnow As Integer

    Private Sub btnExit_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnExit.Click
        Me.Close()

    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnAdd.Click
        If (txtResID.Text <> "" And txtNames.Text <> "" And
txtAddress.Text <> "" And txtBusNo.Text <> "" And
txtGender.Text <> "" And txtType.Text <> "" And
txtDestination.Text <> "" And DateTimePickerTime.Text <> ""
And DateTimePickerDate.Text <> "" And
DateTimePicker1.Text <> "" And txtRates.Text <> "" And
txtAmount.Text <> "" And txtRefundAmount.Text <> "") Then
            Try
```

```
            cn.Close()
            cn.Open()
            Dim query As String
            query = "insert into Cancellation values('" &
Me.txtResID.Text & "','" & Me.txtNames.Text & "','" &
Me.txtAddress.Text & "','" & Me.txtBusNo.Text & "','" &
Me.txtGender.Text & "','" & Me.txtType.Text & "','" &
Me.txtDestination.Text & "','" & Me.DateTimePickerTime.Text
& "','" & Me.DateTimePickerDate.Text & "','" &
Me.DateTimePicker1.Text & "','" & Me.txtRates.Text & "','" &
Me.txtAmount.Text & "','" & Me.txtRefundAmount.Text & "')
delete from Reservation where ResID='" & Me.txtResID.Text
& "' "
            cmd = New SqlCommand(query, cn)
            cmd.ExecuteNonQuery()
            MessageBox.Show("Details are added.", "Details",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
        txtResID.Text = ""
        txtNames.Text = ""
        txtAddress.Text = ""
        txtBusNo.Text = ""
        txtGender.Text = ""
        txtType.Text = ""
        txtDestination.Text = ""
        DateTimePickerDate.Text = ""
        DateTimePickerTime.Text = ""
```

```
            txtRates.Text = ""
            txtAmount.Text = ""
            txtRefundAmount.Text = ""
            txtResID.Enabled = True
            txtNames.Enabled = False
            txtAddress.Enabled = False
            txtBusNo.Enabled = False
            txtGender.Enabled = False
            txtType.Enabled = False
            txtDestination.Enabled = False
            DateTimePickerDate.Enabled = False
            DateTimePickerTime.Enabled = False
            DateTimePicker1.Enabled = False
            txtRates.Enabled = False
            txtAmount.Enabled = False
            txtRefundAmount.Enabled = False
            btnSearch.Enabled = True
            btnAdd.Enabled = False
            btnBack.Enabled = False
            btnAmount.Enabled = False
            Me.txtResID.Focus()
        Else
            MsgBox("PLEASE CLOCK THE AMOUNT
BUTTON...!", MsgBoxStyle.Information +
MsgBoxStyle.OkOnly)
        End If
        cn.Close()

    End Sub
```

```
    Private Sub btnSearch_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnSearch.Click
        If txtResID.Text = "" Then
            MessageBox.Show("PLEASE ENTER THE
RESERVATION ID ...!")
            txtResID.Clear()
        Else
            cn.Close()
            cn.Open()
            Dim query As String
            query = "select
Name,Address,BNo,gen,Type,Destination,Time,Date,TicketRa
tes,amount from reservation where ResID='" & txtResID.Text
& "'"
            cmd = New SqlCommand(query, cn)
            dr = cmd.ExecuteReader
            If dr.HasRows = True Then
                dr.Read()
                Me.txtNames.Text = dr("Name")
                Me.txtAddress.Text = dr("Address")
                Me.txtBusNo.Text = dr("BNo")
                Me.txtGender.Text = dr("gen")
                Me.txtType.Text = dr("Type")
                Me.txtDestination.Text = dr("Destination")
                Me.DateTimePickerDate.Text = dr("Date")
                Me.DateTimePickerTime.Text = dr("Time")
                Me.txtRates.Text = dr("TicketRates")
                Me.txtAmount.Text = dr("Amount")
                btnAmount.Enabled = True
```

```
            dpre = dr("date")
            price = dr("amount")
        Else
            MessageBox.Show("Data not found")
        End If
        cn.Close()
    End If

End Sub

Private Sub Cancellation_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
    txtResID.Text = ""
    txtNames.Text = ""
    txtAddress.Text = ""
    txtBusNo.Text = ""
    txtGender.Text = ""
    txtType.Text = ""
    txtDestination.Text = ""
    DateTimePickerDate.Text = ""
    DateTimePickerTime.Text = ""
    txtRates.Text = ""
    txtAmount.Text = ""
    txtRefundAmount.Text = ""
    txtNames.Enabled = False
    txtAddress.Enabled = False
    txtBusNo.Enabled = False
    txtGender.Enabled = False
    txtType.Enabled = False
```

```
    txtDestination.Enabled = False
    DateTimePickerDate.Enabled = False
    DateTimePickerTime.Enabled = False
    DateTimePicker1.Enabled = False
    txtRates.Enabled = False
    txtAmount.Enabled = False
    txtRefundAmount.Enabled = False
    btnAdd.Enabled = False
    btnBack.Enabled = False
    btnAmount.Enabled = False
    Me.txtResID.Focus()

    Try
        cn.Close()
        cn.Open()
        Dim query As String
        query = "select * from Cancellation"
        cmd = New SqlCommand(query, cn)
        da.SelectCommand = cmd
        da.Fill(ds, "Cancellation")
        dv = New DataView(ds.Tables("Cancellation"))
        Me.DataGridView1.DataSource = dv
        cn.Close()
    Catch ex As Exception
    End Try

End Sub
```

```
    Private Sub txtNames_KeyUp(ByVal sender As Object,
ByVal e As System.Windows.Forms.KeyEventArgs) Handles
txtNames.KeyUp
        If txtNames.Text Like "[0-9]" Or txtNames.Text Like
"*[0-9]*" Or txtNames.Text Like
"[~`@#$%^&*()_+.;:',/?<>}{=!-]" Or txtNames.Text Like
"*[~`@#$%^&*()_+.;:',/?<>}{=!-]*" Then
            MessageBox.Show("Entyer only Alphabets", "Error")
            txtNames.Clear()
        End If

    End Sub

    Private Sub DataGridView1_CellClick1(ByVal sender As
Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs)
Handles DataGridView1.CellClick
        Me.txtResID.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("ResId").Value
        Me.txtNames.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Name").Value
        Me.txtAddress.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Address").Value
        Me.txtBusNo.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("BNo").Value
```

```
        Me.txtGender.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("gen").Value
        Me.txtType.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Type").Value
        Me.txtDestination.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Destination").Value
        Me.DateTimePickerDate.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Date").Value
        Me.DateTimePickerTime.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Time").Value
        Me.DateTimePicker1.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Today").Value
        Me.txtRates.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("TicketRates").Value
        Me.txtAmount.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Amount").Value
        Me.txtRefundAmount.Text =
Me.DataGridView1.Rows.Item(Me.DataGridView1.CurrentRo
w.Index()).Cells.Item("Refund").Value

    End Sub
```

```
    Private Sub btnAmount_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnAmount.Click
        'no = 1
        dnow = dpre.Day - Me.DateTimePicker1.Value.Day
        If (txtResID.Text <> "" And txtNames.Text <> "" And
txtAddress.Text <> "" And txtBusNo.Text <> "" And
txtGender.Text <> "" And txtType.Text <> "" And
txtDestination.Text <> "" And DateTimePickerTime.Text <> ""
And DateTimePickerDate.Text <> "" And
DateTimePicker1.Text <> "" And txtRates.Text <> "" And
txtAmount.Text <> "") Then
            If (dnow <= 0) Then
                MessageBox.Show("No Deduction")
                btnAmount.Enabled = True
                btnAdd.Enabled = False
                btnBack.Enabled = True
            Else
                If (dnow >= 2) Then
                    money = price - ((15 * price) / 100)
                    Me.txtRefundAmount.Text = money
                    btnAmount.Enabled = False
                    btnAdd.Enabled = True
                Else
                    MessageBox.Show("No Deduction")
                    btnAmount.Enabled = True
                    btnAdd.Enabled = False
                    btnBack.Enabled = True
                End If
            End If
```

```
        Else
            MsgBox("PLEASE ENSURE THAT ALL FIELDS
ARE COMPLETE....!", MsgBoxStyle.Information +
MsgBoxStyle.OkOnly)
            btnAmount.Enabled = True
            btnAdd.Enabled = False
            btnBack.Enabled = True
        End If

    End Sub

    Private Sub btnBack_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnBack.Click
        txtResID.Text = ""
        txtNames.Text = ""
        txtAddress.Text = ""
        txtBusNo.Text = ""
        txtGender.Text = ""
        txtType.Text = ""
        txtDestination.Text = ""
        DateTimePickerDate.Text = ""
        DateTimePickerTime.Text = ""
        txtRates.Text = ""
        txtAmount.Text = ""
        txtRefundAmount.Text = ""
        btnSearch.Enabled = True
        btnAdd.Enabled = False
        btnBack.Enabled = False
        btnAmount.Enabled = False
        Me.txtResID.Focus()
```

```
    End Sub

End Class
```

TESTING

Testing is very vital for any system to be successfully implemented. The common view is that it is performed to prove that there are no errors in a program. Therefore the most useful and practical approach is with the explicit intention of finding the errors. The system is tested experimentally to ensure that the software does not fail. The system is run according to its specifications and in the way the user expects. Following testing practices are used. The system will process as normal input preparation of test-sample data.

STRATERGIES FOR TESTING

Unit Testing

Each and every module was intensively tested to check for errors and defects. All possible mistakes were rectified. Manually code is tested like logical errors.

Once the manual checking is over the compilation has been done. Syntactical error if any has to be corrected.

After the clean compilation of the program, some dummy data as per specifications has been used for testing of that module to see if it works as specified.

Integration Testing

Integration testing uncovers errors that arise when modules are integrated to build the overall system.

The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages)

All the unit tested modules were integrated & the errors that occurred were removed and the overall program structure was build as specified by the design.

System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

System testing is used to detect defects both within the "inter-assemblages" and also within the system as a whole.

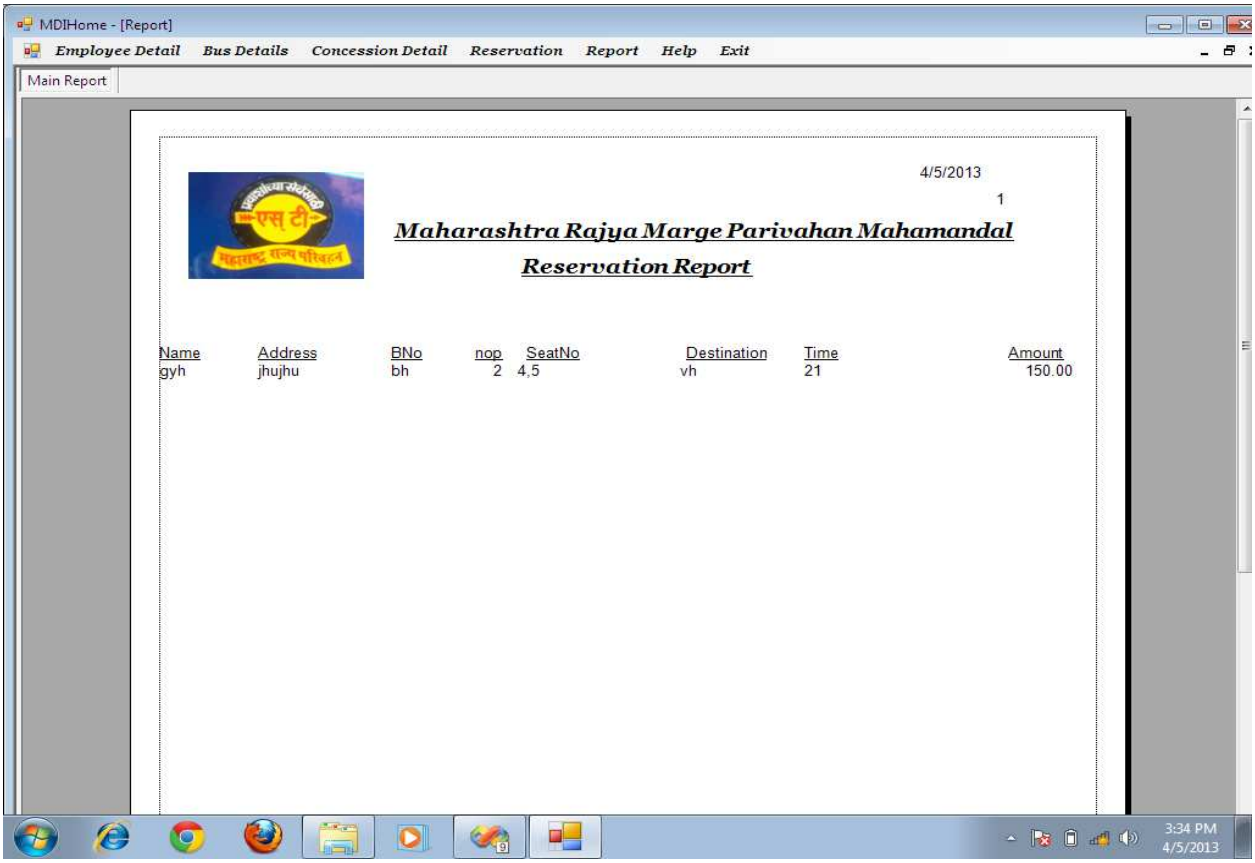
TEST CASES

Test Case Id	Test Case Name	Test Case Description	Step	Executed Result	Actual	Test Case Status	Priority
Login	Validate login	To verify appropriate user login	Enter number or special characters	Invalid Username or password	Error Message	Designed	High
Driver Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium
Employee Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium
Conductor Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium
Reservation Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium
Cancellation Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium
Concession Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium

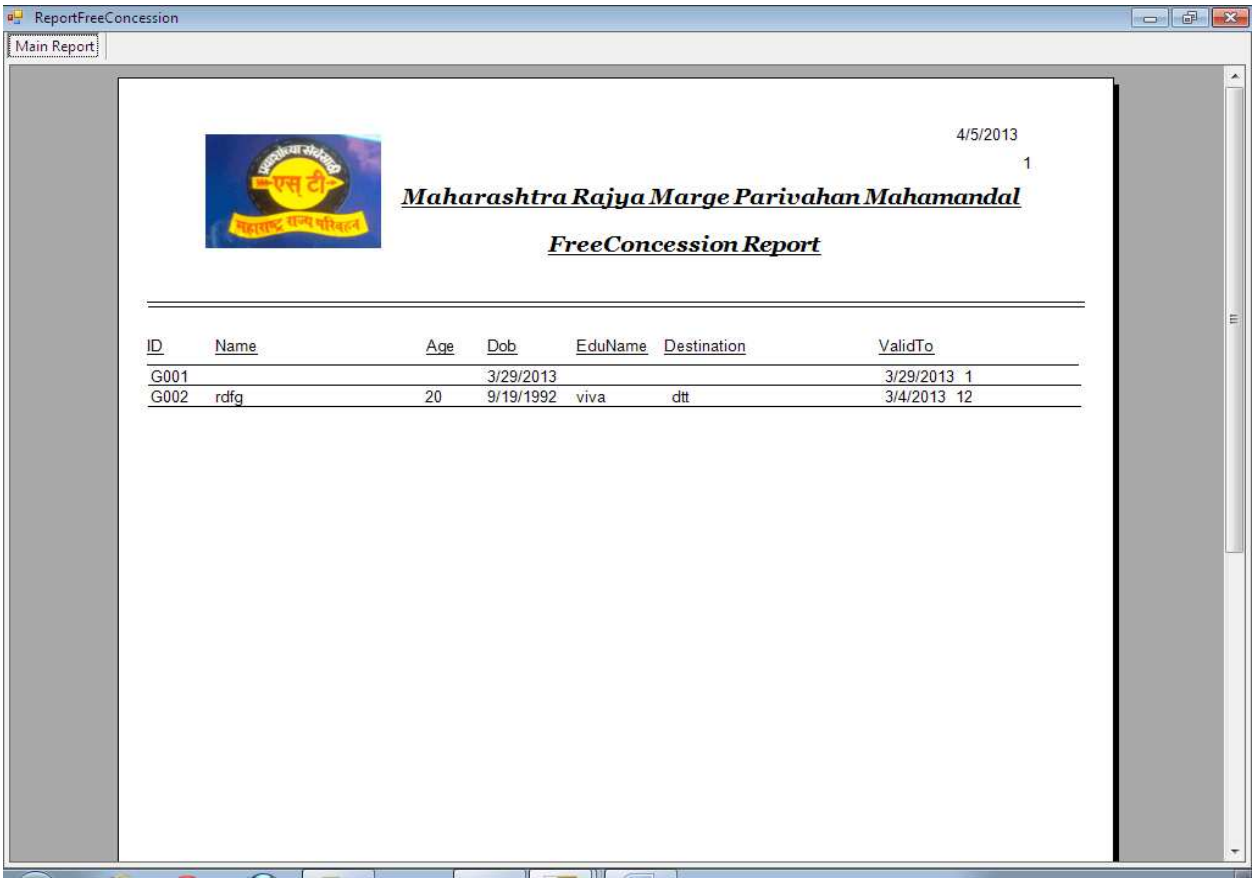
Bus Master Details	Validate fields	To verify all the fields are filed	Leave any field empty	Add & Update button disabled	Button disabled	Designed	Medium

Reports

Reservation



Concession Report



Code

```
Imports System.Collections
Imports System.Data.SqlClient
Imports System.Data.OleDb

Public Class Reportconcession
    Dim rpt As New CrystalReportconcession
    Dim selectquery As New SqlCommand
    Dim myDA As New SqlDataAdapter()
    Public sqlConnect As New SqlConnection("Data
Source=.;Initial Catalog=BusSystemManagement;Integrated
Security=True")
    Dim myDS As New DataSet()
    Private Sub Report_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        personal()
    End Sub

    Public Function personal() As Boolean
        Me.CrystalReportViewer1.RefreshReport()
        Dim query As String = "SELECT *  FROM Normal"
        selectquery = New SqlCommand(query, sqlConnect)
        myDA.SelectCommand = selectquery
        myDA.Fill(myDS, "Normal")
        rpt.SetDataSource(myDS)
        CrystalReportViewer1.ReportSource = rpt
        Me.CrystalReportViewer1.RefreshReport()
    End Function
```

```
    Private Sub CrystalReportViewer1_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CrystalReportViewer1.Load

    End Sub
End Class
```

FUTURE ENHNACEMENT

- Logical and technical error will be less found in future.

- System will be more user friendly as compare to today's system.

- System will have more detail information about the different places and other new routes.

- In the security vision the user name and password authentication is more secure than today's model.

PROCEDURE OF DEPLOYMENT

☞Install the Microsoft SQL 2008.

☞Restore or attach the database backup.

☞Install the VB.NET .

☞Establish the connection with local SQL server

☞Double Click on the setup file.

CONCLUSION

It was great opportunity for us as a student to learn and understand various aspects associated with project development. I did undergo from various phases of project development life cycle like analysis , design ,coding , implementation , and testing.

The preceding material is a sincere effort from my side to create the “STATE TRANSPORT MANAGEMENT” software as my project work for BSc.I.T project . I got the idea about the ups and downs taking place during the project development. I analyzed the problems and solved those problems that were faced in my project.

The project shows the flow of each and every transaction which is being carried out by the desired user successfully thus giving him the desired result.

BIBLIOGRAPHY

- System Analysis And Design In A Changing World
  - Satzinger, Jackson and Burd
- VB.NET
- SQL 2 - James Groff
- Software Engineering – A Parishioners Approach
  - Roger S. Pressman