**ABSTRACT**

This System is used to maintain the bus pass for students .With this system we can reduce the multiple steps involved in creating students bus pass. It reduces the time taken for creating a student bus pass.It helps in reducing the chances of manipulating the data . It reduce the need of students physical presence at bus pass counter.The created bus pass ID of a student is based on the College ID. If the student loses the bus pass, the new one is issued without any delay. In this system the RTC clerk can view all the reports like College list, Students list and Expired Tickets List etc.

The College List is used for knowing the college details like college name, College phone number etc .The Students List is used for verifying the student is valid or not. The Expired Tickets report is used for identifying the Expired Tickets Details in Monthly wise. In this system the RTC clerk can Generate the ID card based on Bus pass ID.In the present days it is very difficult maintain the students bus pass. So This Project “STUDENTS BUS PASS MANAGEMENT SYSTEM” is helpful to maintain the Bus pass System.

**2.1 Introduction to Project**

The Project “Students Bus pass Management system” is helpful to reduce the multiple steps involved in creating students bus pass. It reduces the time taken for creating a student bus pass.It helps in reducing the chances of manipulating the data . It reduce the need of students physical presence at bus pass counter.The created bus pass ID of a student is based on the College ID. If the student loses the bus pass, the new one is issued without any delay. In this system the RTC clerk can view all the reports like College list, Students list and Expired Tickets List etc.

**2.2 Existing System**

In the Existing system multiple steps are involved in creating students bus pass ex: Verification, Id card generation and finally Ticket will be generated. The generated Bus pass ID is Complex, which is not easy for a student to remember.If the students lose the bus pass, a new pass is not easily issued. They need to reapply as fresher for student bus pass.

**Disadvantages**

* + Multiple Steps involved in creation of Bus pass .
  + Generated ID is Complex to remember.

1. **Proposed System**

In the Proposed System we can reduce the multiple steps in the process of creating a new Bus pass. It helps in reducing the chances of manipulating the data. The generated Bus pass ID is similar to student’s Roll Id which is easy remember ex:S16PVPMCA101.If the students lose the bus pass, a new pass is issued easily based on Student bus pass ID.We can find all expired bus pass tickets and also the students details.

**Advantages**

* This system reduce the multiple steps in the process of creating a new Bus pass.
* The generated Bus pass ID is similar to student’s Roll Id which is easy remember ex:S16PVPMCA101.
* We can find all expired bus pass tickets and also the students details

**3.1 REQUIREMENTS SPECIFIACTION AND ANALYSIS**

**Software Requirements Specifications**

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to the software as a part of system.

Engineering are refined by establishing a complete information description, a detailed functional and behavioral description, and indication of performance requirements and design constraints, appropriate validation criteria and other data pertinent to requirements. An outline of the Software Requirements Specification:

**3.2 FEASIBILITY STUDY**

All projects are feasible, given unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Three key considerations are involved in the feasibility analysis.

**Economic Feasibility**

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

**Technical Feasibility**

Technical feasibility centers on the existing computer system (hardware, software, etc.,) and to what extent it can support the proposed addition. If the budget is a serious constraint, then the project is judged not feasible.

**Operational Feasibility**

People are inherently resistant to change, and computers have been known to facilitate change. It is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the staff on new ways of conducting business.

Software system requirements are often classified as functional requirements and non-functional requirements.

1. **Functional Requirements:**

These are the statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

1. **Non-functional Requirements:**

These are the constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services.

**FUNCTIONAL REQUIREMENTS :**

The functional requirements are the critical requirements which should be within the system in order to overcome or at least minimize the drawbacks of the existing system.

Functional requirements comprise of what are and what type of data should be given as input to the system, what processing should be done to the inputs in order to get the required working, functionalities and output as expected by the user and in what ways or how the output must be presented to the user.

**NON FUNCTIONAL REQUIREMENTS:**

The non-functional requirements are those that do not directly contribute to the system functionality, modules or processing but are major factors contributing to the quality of the system.

**Portability:** The software can be installed on all necessary platforms on which it is expected to run.

**Performance:** Performance is to estimate whether it is possible to develop the proposed system with the available hardware, software and network resources. The performance is evaluated on basis of factor like ease of development, availability of hardware and re-usable code availability. Since all the hardware, software and network requirements are easily available; the development of application became feasible.

**Efficiency**: Specifies how well the software utilizes scarce resources: CPU cycles, disk space,memory, bandwidth etc. All of the above mentioned resources can be effectively used by performing most of the validations at client side and reducing the workload on server.

**Flexibility**: If the organization intends to increase or extend the functionality of the softwareafter it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules.

**3.3 SYSTEM REQUIREMENTS**

**SOFTWARE REQUIREMENTS**

* Database : ORACLE10G.
* Tools : Visual Studio.
* Operating System : Windows 7.

**HARDWARE REQUIREMENTS**

* Processor : Intel PIV
* RAM : 2GB RAM
* Hard disk : 2GB or more

**4.1MODULES**

1. Students Verification .
2. Colleges Data Entry.
3. Bus Pass ID card And Ticket Generation.
4. Students Data Entry.
5. Bus Pass Data Entry.
6. Expired Tickets Details.

**Module Description**

**1.Students Verification**: This module is used for Verifying the students details.

**2Colleges Data Entry** : This module is used for maintaining the College details.

**3.Bus Pass ID card And Ticket Generation**: This is used for generating Bus Pass ID card and Ticket.

**4.Student Data Entry** : This is helpful for maintaining the students details.

**5. Bus Pass Data Entry**: This is helpful for entering and maintaining the students bus pass data.

**6.Expired Tickets Details**: This is helpful for viewing the Expired Tickets

**4.2 UML DIAGRAMS**

**CONTENTS:**

1. Class Diagram
2. Use Case Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Activity Diagram
6. State Chart Diagram
7. Component Diagram
8. Deployment Diagram
9. This application represents all the activities in a Students Bus Pass Management System.
10. It Shows the Structure and behavior of the Students Bus Pass Management System with the help of class and interaction diagrams.
11. It shows all the responsibilities of a Bus Pass Clerk, College and Students.
12. It also maintains the activities of Information System and College details, Students details etc.

**1.Class Diagram:**

A **class diagram** shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems.

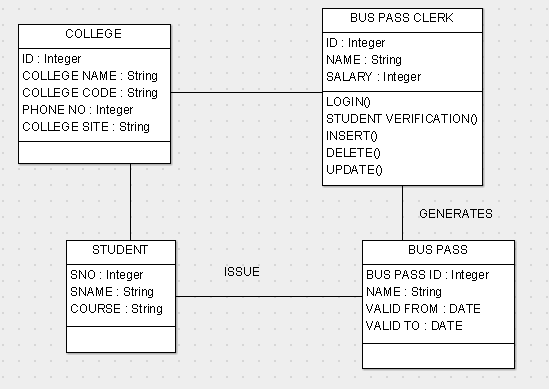
From The Conceptualization Analysis the Classes Are

 COLLEGE

STUDENT

 BUS PASS

 BUS PASS CLERK



**2.UseCase Diagram:**

An **Usecase diagram** shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system.

This diagram shows how the COLLEGE CLERK, BUS PASS CLERK and STUDENT involves in the STUDENTS BUS PASS MANAGEMENT SYSTEM

For Students Bus Pass Management System we use following use cases

 VERIFYING OF STUDENT

 SUBMISSION OF STUDENTS DETAILS

 ISSUE OF BUS PASS

 ISSUE OF BUS PASS TICKET

 CREATION OF NEW BUS PASS

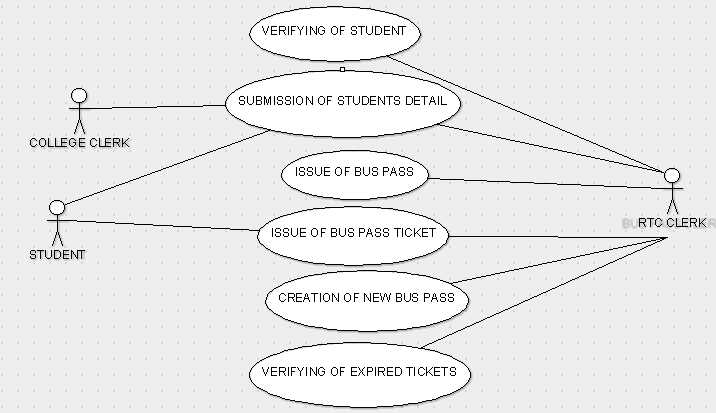
 VERIFYING OF EXPIRED TICKETS

There are two actors involved in the system

COLLEGE CLERK

 BUS PASS CLERK

 STUDENT



**3.Sequence Diagram:**

A **Sequence diagram** is an interaction diagram that emphasizes the time-ordering of messages; a communication diagram is an interaction diagram that emphasizes the structural organization of the objects or roles that send and receive messages.

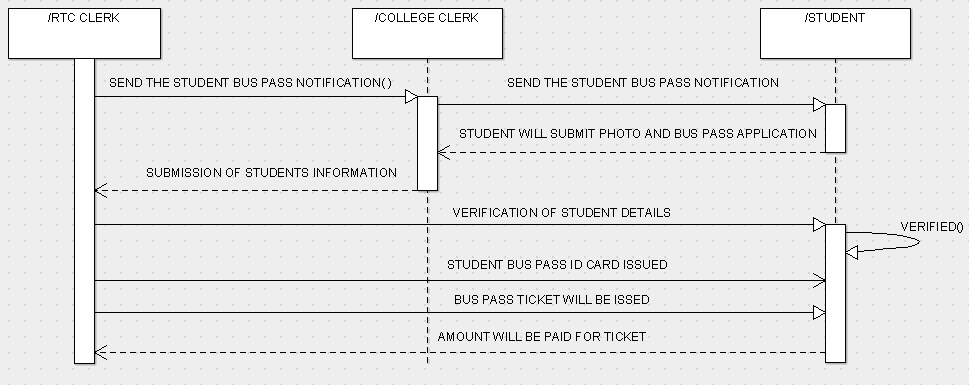
 The SEQUENCE DIAGRAM is FOR CREATING A NEW BUS PASS.

The objects involved in this system are

COLLEGE CLERK

 BUS PASS CLERK

 STUDENT



**4.Collaboration Diagram:**

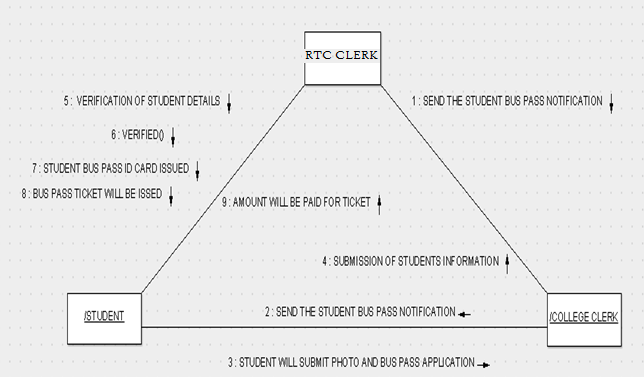
 The COLLABORATION DIAGRAMis FOR CREATING A NEW BUS PASS

The objects involved in this system are

COLLEGE CLERK

 BUS PASS CLERK

 STUDENT



**5.Activity Diagram:**

An **Activity diagram** shows the structure of a process or other computation as the flow of control and data from step to step within the computation.

The Following Actions are used in the Activity Diagram

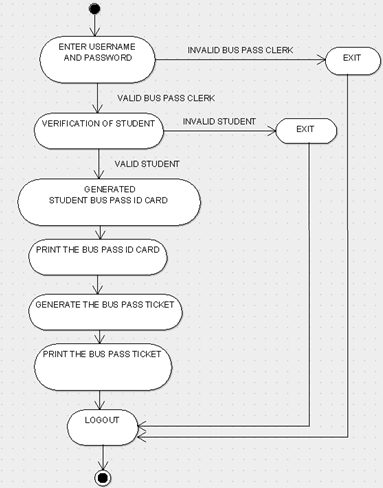
 VERIFICATION OF STUDENT

 GENERATE STUDENT BUS PASS ID CARD

 PRINT THE BUS PASS ID CARD

 GENERATE THE BUS PASS TICKET

 PRINT THE TICKET



**6.State Chart Diagram:**

A **state diagram** shows a state machine, consisting of states, transitions, events, and activities.

The Following States Are Used In The State Chart Diagram

 VALIDATION

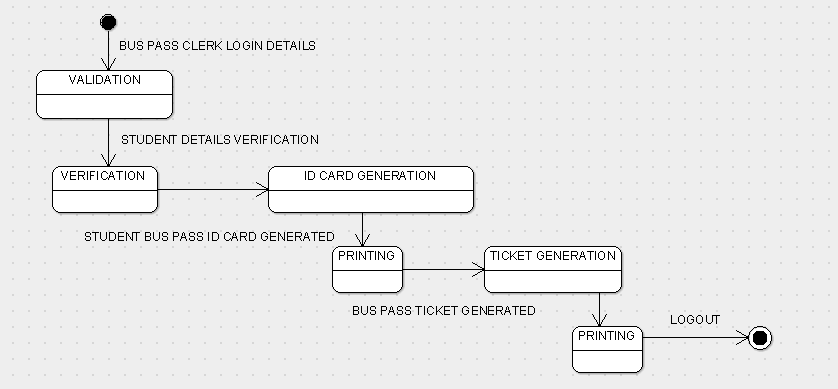
 VERIFICATION

 ID CARD GENERATION

 PRINTING

 TICKET GENERATION

 PRINTING



**7.Component Diagram:**

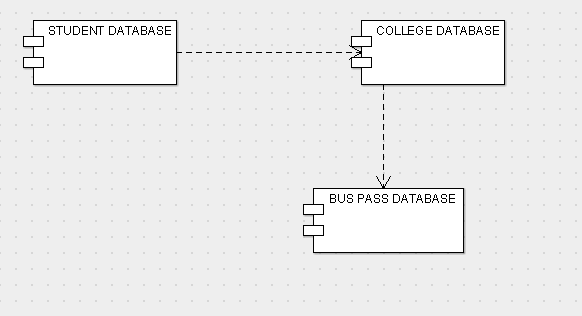
A **Component Diagram** shows the configuration of run-time processing nodes and the components that live on them.

The Following Components Are Used In The Component Diagram

COLLEGE DATABASE

 BUS PASS DATABASE

 STUDENT DATABASE



**8.Deployment Diagram:**

**Deployment diagrams** address the static deployment view of architecture. A node typically hosts one or more artifacts.

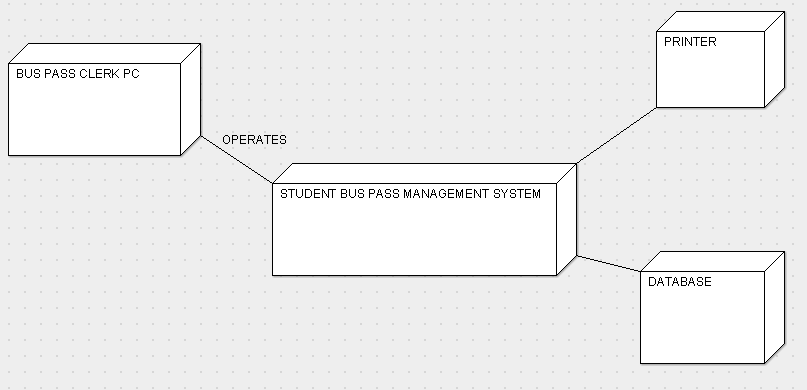
The Following are used In The Deployment Diagram

 BUS PASS CLERK PC

 STUDENT BUS PASS MANAGEMENT SYSTEM

 PRINTER

 DATABASE



**4.3DATABASES**

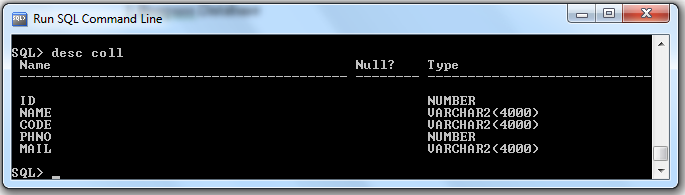
In this project there are three relations in Database they are

1.College Relation

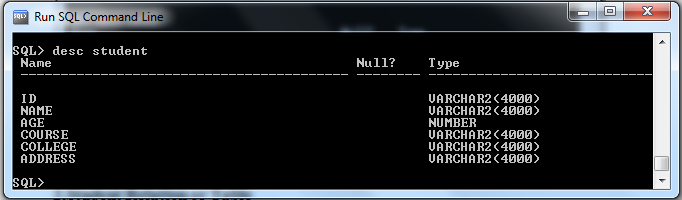
2.Student Relation

3.Buspass Relation

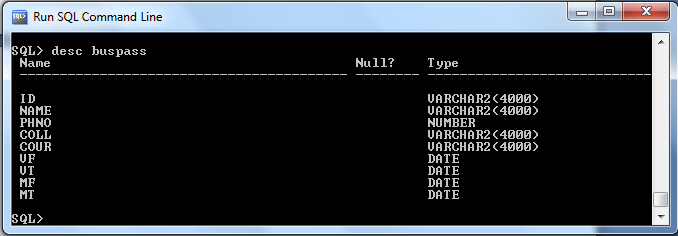
**1.College Relation or Table**

****

**2.Student Relation or Table**

****

**3.Buspass Relation or Table**

****

**5.1 SAMPLE CODE:**

**Login.vb**

PublicClass Form1

PrivateSub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

'Login()

If TextBox1.Text = "hussain"And TextBox2.Text = "pvpsit"Then

Form17.Show()

Me.Hide()

Else

MsgBox("wrong password", MsgBoxStyle.Critical)

EndIf

EndSub

PrivateSub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

'Clear

TextBox1.Text = ""

TextBox2.Text = ""

TextBox1.Focus()

EndSub

PrivateSub PictureBox3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox3.Click

Me.Close()

EndSub

EndClass

**College.Vb**

Imports System.Data.OleDb

PublicClass Form3

Dim con AsNew OleDbConnection

Dim dr As OleDbDataReader

PrivateSub Form3\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

con.ConnectionString = "Provider=MSDAORA.1;Password=oracle;User ID=system;Persist Security Info=True"

con.Open()

MsgBox("connected")

Dim cmd AsNew OleDbCommand("select \* from COLL", con)

dr = cmd.ExecuteReader()

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

EndIf

EndSub

PrivateSub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

'NEXT

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

Else

MsgBox("last record")

EndIf

EndSub

PrivateSub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

'INSERT

Dim cmd AsNew OleDbCommand("insert into COLL values("& Val(TextBox1.Text) &",'"& (TextBox2.Text) &"','"& (TextBox3.Text) &"',"& Val(TextBox4.Text) &",'"& (TextBox5.Text) &"')", con)

cmd.ExecuteNonQuery()

MsgBox("record Inserted ")

EndSub

PrivateSub PictureBox3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox3.Click

'DELETE

Dim n AsInteger

n = InputBox("Enter COLLEGE No:")

Dim cmd AsNew OleDbCommand(" delete from COLL where ID="& n, con)

cmd.ExecuteNonQuery()

MsgBox("Record Deleted ")

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

TextBox5.Text = ""

EndSub

PrivateSub PictureBox7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click

'SEARCH

Dim n AsInteger

n = InputBox("enter COLLEGE NO : ")

Dim cmd AsNew OleDbCommand(" Select \* from COLL where ID="& n, con)

dr = cmd.ExecuteReader()

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

Else

MsgBox(" No such record found ")

EndIf

EndSub

PrivateSub PictureBox6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox6.Click

'UPDATE

Dim cmd AsNew OleDbCommand("update COLL set NAME = '"& (TextBox2.Text) &" ',CODE = '"& (TextBox3.Text) &"', PHNO = '"& Val(TextBox4.Text) &"', MAIL = '"& (TextBox5.Text) &"' where ID = "& Val(TextBox1.Text), con)

cmd.ExecuteNonQuery()

MsgBox(" Record UPDATED ")

EndSub

PrivateSub PictureBox5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox5.Click

'CLEAR

TextBox1.Text = " "

TextBox2.Text = " "

TextBox3.Text = " "

TextBox4.Text = " "

TextBox5.Text = " "

TextBox1.Focus()

EndSub

PrivateSub PictureBox4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox4.Click

Form2.Show()

Me.Close()

EndSub

EndClass

**List of College.vb**

Imports System.Data.OleDb

PublicClass Form4

Dim con AsNew OleDbConnection

Dim dr As OleDbDataReader

PrivateSub Form4\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

con.ConnectionString = "Provider=MSDAORA.1;Password=oracle;User ID=system;Persist Security Info=True"

con.Open()

MsgBox("connected")

Dim cmd AsNew OleDbCommand("select \* from COLL", con)

dr = cmd.ExecuteReader()

While dr.Read

ListBox1.Items.Add(dr.GetValue(0))

ListBox2.Items.Add(dr.GetValue(1))

ListBox3.Items.Add(dr.GetValue(2))

ListBox4.Items.Add(dr.GetValue(3))

ListBox5.Items.Add(dr.GetValue(4))

EndWhile

EndSub

PrivateSub PictureBox7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click

EndSub

PrivateSub PictureBox4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox4.Click

Form2.Show()

Me.Close()

EndSub

EndClass Imports System.Data.OleDb

PublicClass Form4

Dim con AsNew OleDbConnection

Dim dr As OleDbDataReader

PrivateSub Form4\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

con.ConnectionString = "Provider=MSDAORA.1;Password=oracle;User ID=system;Persist Security Info=True"

con.Open()

MsgBox("connected")

Dim cmd AsNew OleDbCommand("select \* from COLL", con)

dr = cmd.ExecuteReader()

While dr.Read

ListBox1.Items.Add(dr.GetValue(0))

ListBox2.Items.Add(dr.GetValue(1))

ListBox3.Items.Add(dr.GetValue(2))

ListBox4.Items.Add(dr.GetValue(3))

ListBox5.Items.Add(dr.GetValue(4))

EndWhile

EndSub

PrivateSub PictureBox7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click

EndSub

PrivateSub PictureBox4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox4.Click

Form2.Show()

Me.Close()

EndSub

EndClass

**Student.vb**

Imports System.Data.OleDb

PublicClass Form5

Dim con AsNew OleDbConnection

Dim dr As OleDbDataReader

PrivateSub Form5\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

con.ConnectionString = "Provider=MSDAORA.1;Password=oracle;User ID=system;Persist Security Info=True"

con.Open()

MsgBox("connected")

Dim cmd AsNew OleDbCommand("select \* from STUDENT", con)

dr = cmd.ExecuteReader()

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

TextBox6.Text = dr.GetValue(5)

EndIf

EndSub

PrivateSub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

'NEXT

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

TextBox6.Text = dr.GetValue(5)

Else

MsgBox("last record")

EndIf

EndSub

PrivateSub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

'INSERT

Dim cmd AsNew OleDbCommand("insert into STUDENT values('"& (TextBox1.Text) &"','"& (TextBox2.Text) &"',"& Val(TextBox3.Text) &",'"& (TextBox4.Text) &"','"& (TextBox5.Text) &"','"& (TextBox6.Text) &"')", con)

cmd.ExecuteNonQuery()

MsgBox("record Inserted ")

EndSub

PrivateSub PictureBox7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click

'SEARCH

Dim cmd AsNew OleDbCommand(" Select \* from STUDENT where ID LIKE'%"& TextBox7.Text &"%'", con)

dr = cmd.ExecuteReader()

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

TextBox6.Text = dr.GetValue(5)

Else

MsgBox(" No such record found ")

EndIf

EndSub

PrivateSub PictureBox6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox6.Click

'DELETE

Dim cmd AsNew OleDbCommand(" DELETE STUDENT where ID LIKE'%"& TextBox7.Text &"%'", con)

cmd.ExecuteNonQuery()

MsgBox("Record Deleted ")

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ""

TextBox5.Text = ""

TextBox6.Text = ""

EndSub

PrivateSub PictureBox3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox3.Click

'UPDATE

Dim cmd AsNew OleDbCommand("update STUDENT set NAME = '"& (TextBox2.Text) &" ',AGE = "& Val(TextBox3.Text) &", COURSE = '"& (TextBox4.Text) &"', COLLEGE = '"& (TextBox5.Text) &"',ADDRESS='"& (TextBox6.Text) &"' where ID = '"& (TextBox1.Text) &"'", con)

cmd.ExecuteNonQuery()

MsgBox(" Record UPDATED ")

EndSub

PrivateSub PictureBox4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox4.Click

'CLEAR

TextBox1.Text = " "

TextBox2.Text = " "

TextBox3.Text = " "

TextBox4.Text = " "

TextBox5.Text = " "

TextBox6.Text = ""

TextBox1.Focus()

EndSub

**Bus pass.vb**

Imports System.Data.OleDb

PublicClass Form6

Dim con AsNew OleDbConnection

Dim dr As OleDbDataReader

PrivateSub Form6\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

con.ConnectionString = "Provider=MSDAORA.1;Password=oracle;User ID=system;Persist Security Info=True"

con.Open()

MsgBox("connected")

Dim cmd AsNew OleDbCommand("select \* from BUSPASS", con)

dr = cmd.ExecuteReader()

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

TextBox6.Text = dr.GetValue(5)

TextBox7.Text = dr.GetValue(6)

TextBox8.Text = dr.GetValue(7)

TextBox9.Text = dr.GetValue(8)

EndIf

EndSub

PrivateSub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

'NEXT

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

TextBox6.Text = dr.GetValue(5)

TextBox7.Text = dr.GetValue(6)

TextBox8.Text = dr.GetValue(7)

TextBox9.Text = dr.GetValue(8)

Else

MsgBox("last record")

EndIf

EndSub

PrivateSub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

'INSERT

Dim cmd AsNew OleDbCommand("insert into BUSPASS values('"& (TextBox1.Text) &"','"& (TextBox2.Text) &"',"& Val(TextBox3.Text) &",'"& (TextBox4.Text) &"','"& (TextBox5.Text) &"','"& (TextBox6.Text) &"','"& (TextBox7.Text) &"','"& (TextBox8.Text) &"','"& (TextBox9.Text) &"')", con)

cmd.ExecuteNonQuery()

MsgBox("record Inserted ")

EndSub

PrivateSub PictureBox4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox4.Click

'CLEAR

TextBox1.Text = " "

TextBox2.Text = " "

TextBox3.Text = " "

TextBox4.Text = " "

TextBox5.Text = " "

TextBox6.Text = " "

TextBox7.Text = " "

TextBox8.Text = " "

TextBox9.Text = ""

TextBox1.Focus()

EndSub

PrivateSub PictureBox7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click

'SEARCH

Dim cmd AsNew OleDbCommand(" Select \* from BUSPASS where ID LIKE'%"& TextBox10.Text &"%'", con)

dr = cmd.ExecuteReader()

If dr.Read Then

TextBox1.Text = dr.GetValue(0)

TextBox2.Text = dr.GetValue(1)

TextBox3.Text = dr.GetValue(2)

TextBox4.Text = dr.GetValue(3)

TextBox5.Text = dr.GetValue(4)

TextBox6.Text = dr.GetValue(5)

TextBox7.Text = dr.GetValue(6)

TextBox8.Text = dr.GetValue(7)

TextBox9.Text = dr.GetValue(8)

Else

MsgBox(" No such record found ")

EndIf

EndSub

PrivateSub PictureBox6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox6.Click

'DELETE

Dim cmd AsNew OleDbCommand(" DELETE BUSPASS where ID LIKE'%"& TextBox10.Text &"%'", con)

cmd.ExecuteNonQuery()

MsgBox("Record Deleted ")

TextBox1.Text = " "

TextBox2.Text = " "

TextBox3.Text = " "

TextBox4.Text = " "

TextBox5.Text = " "

TextBox6.Text = " "

TextBox7.Text = " "

TextBox8.Text = " "

TextBox9.Text = ""

TextBox1.Focus()

EndSub

PrivateSub PictureBox3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox3.Click

'UPDATE

Dim cmd AsNew OleDbCommand("update BUSPASS set NAME = '"& (TextBox2.Text) &" ',PHNO = "& Val(TextBox3.Text) &", COLL = '"& (TextBox4.Text) &"', COUR = '"& (TextBox5.Text) &"',VF ='"& (TextBox6.Text) &"',VT = '"& (TextBox7.Text) &"',MF = '"& (TextBox8.Text) &"',MT = '"& (TextBox9.Text) &"' where ID = '"& (TextBox1.Text) &"'", con)

cmd.ExecuteNonQuery()

MsgBox(" Record UPDATED ")

EndSub

PrivateSub PictureBox5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox5.Click

Form17.Show()

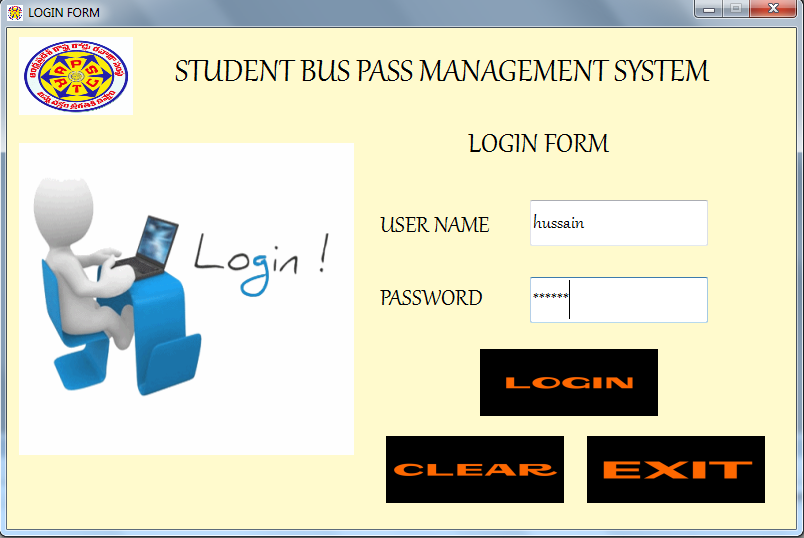
Me.Hide()

EndSub

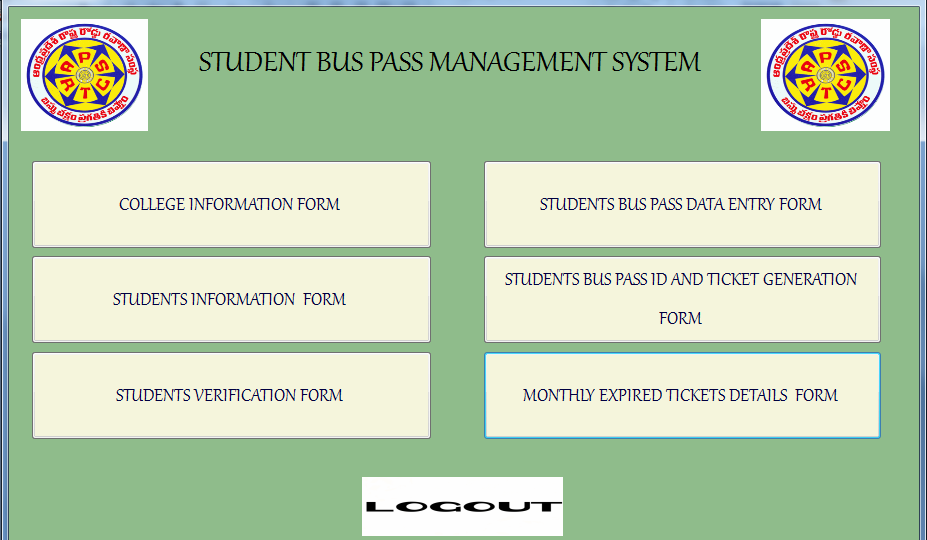
EndClass

**5.2SCREEN SHOTS**

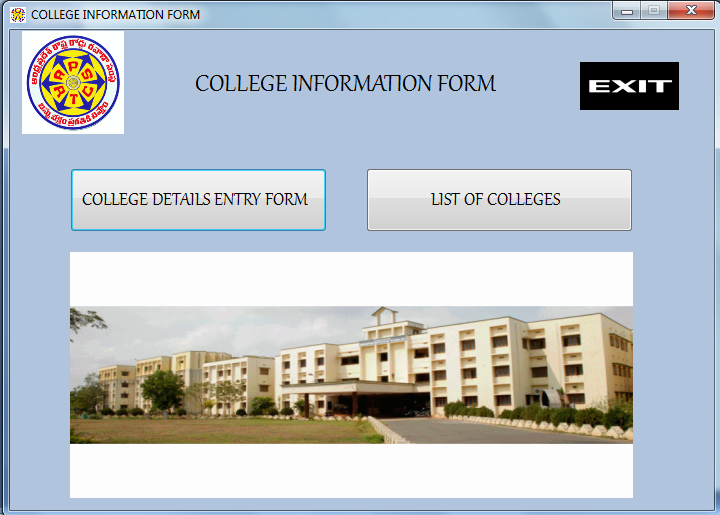
**LOGIN SCREEN**

****

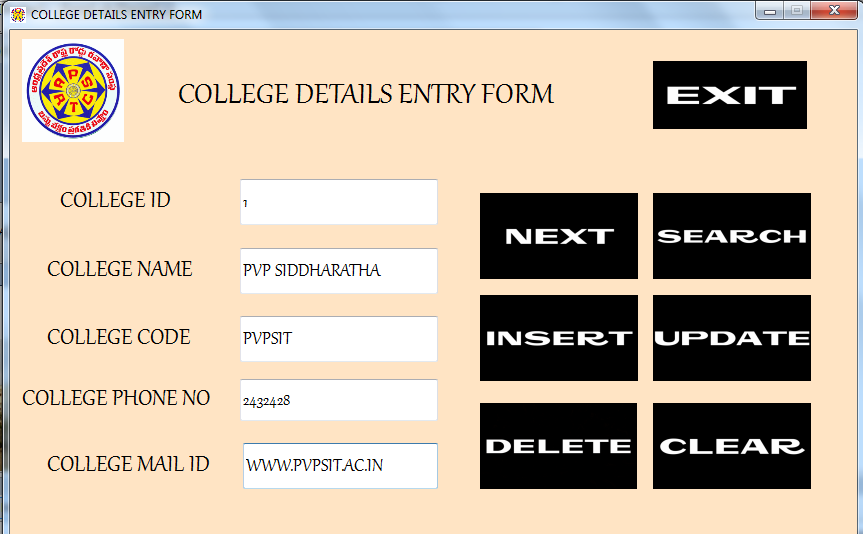
**MAIN SCREEN**

****

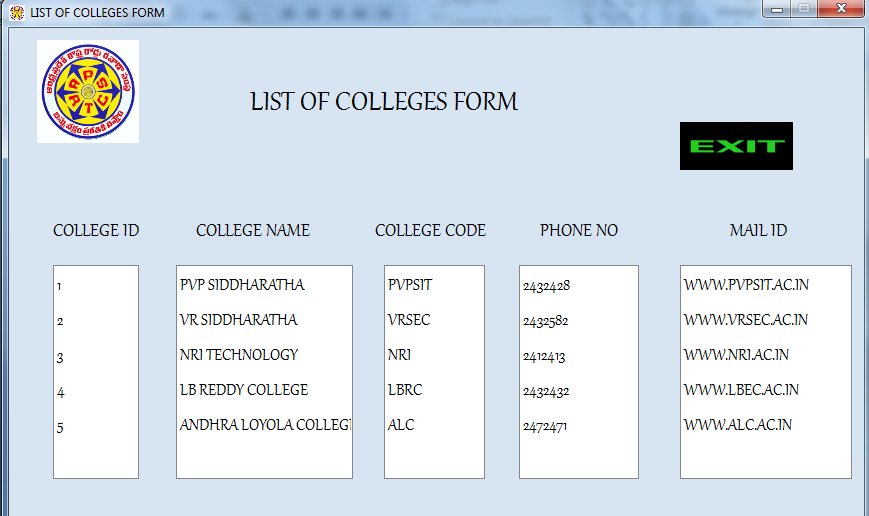
**COLLEGE INFORMATION SCREEN**

****

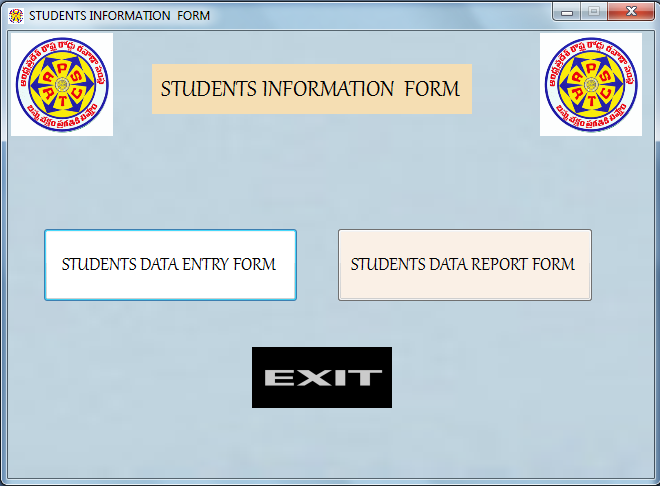
**COLLEGE DATA ENTRY SCREEN**

****

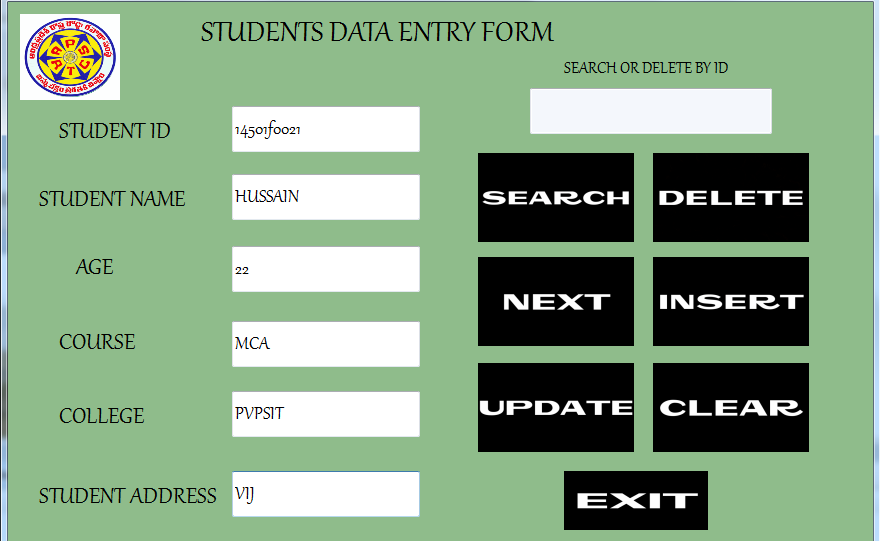
**LIST OF COLLEGE SCREEN**

****

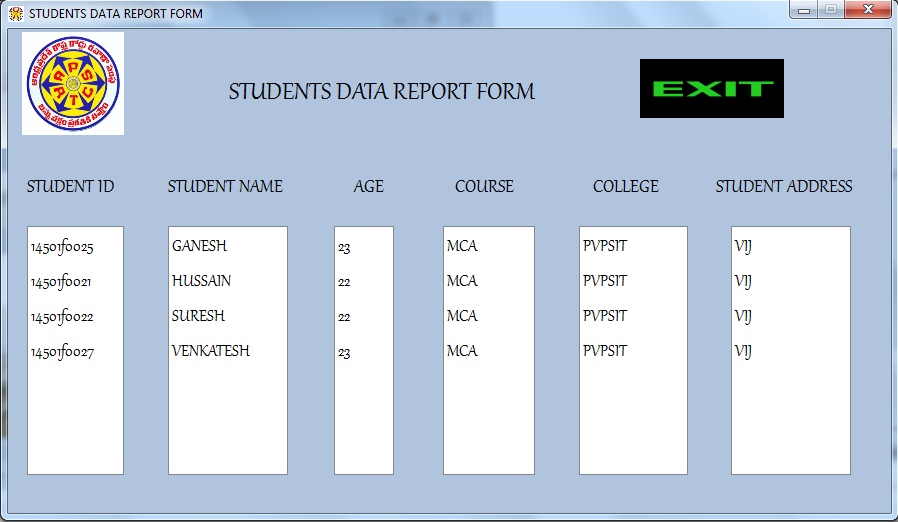
**STUDENTS INFORMATION SCREEN**

****

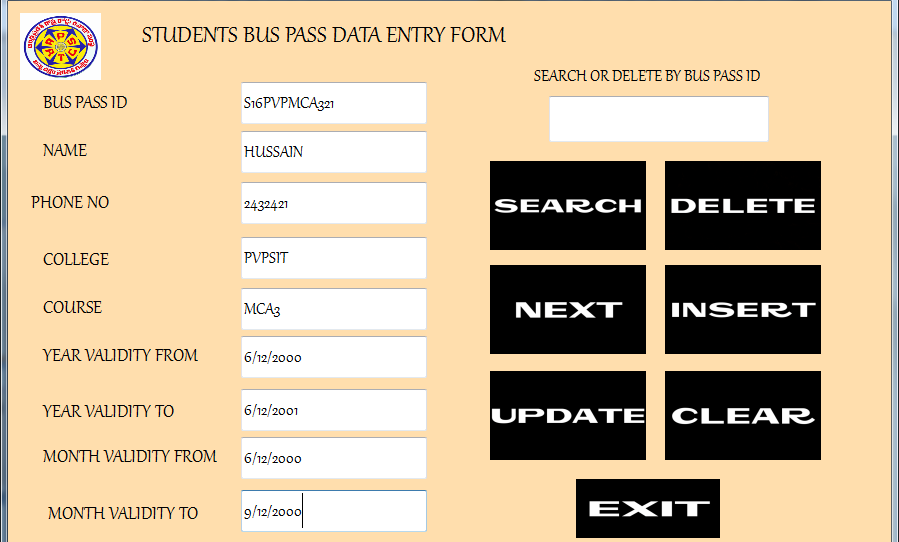
**STUDENTS DATA ENTRY SCREEN**

****

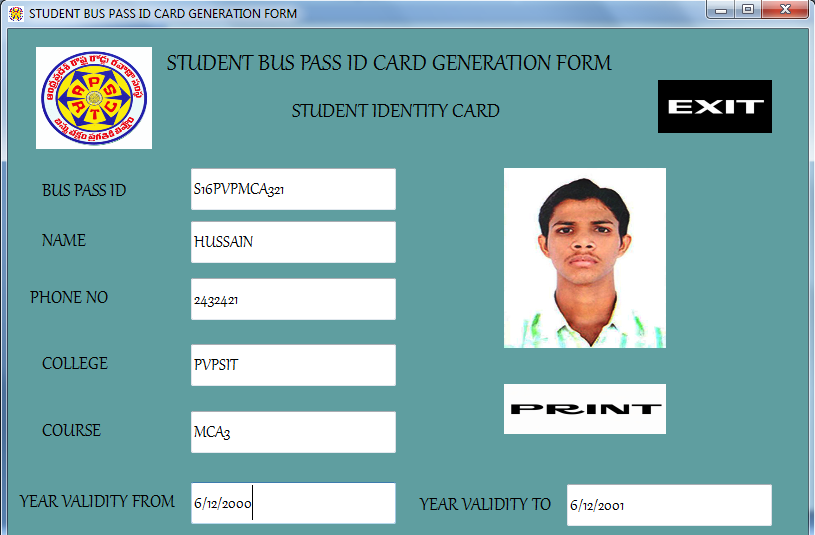
**STUDENTS DATA REPORT SCREEN**

****

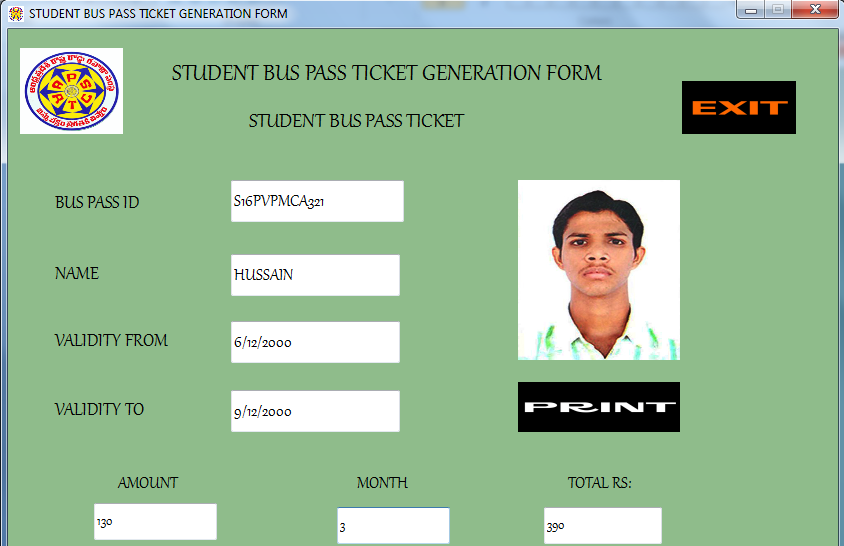
**BUS PASS DATA ENTRY SCREEN**

****

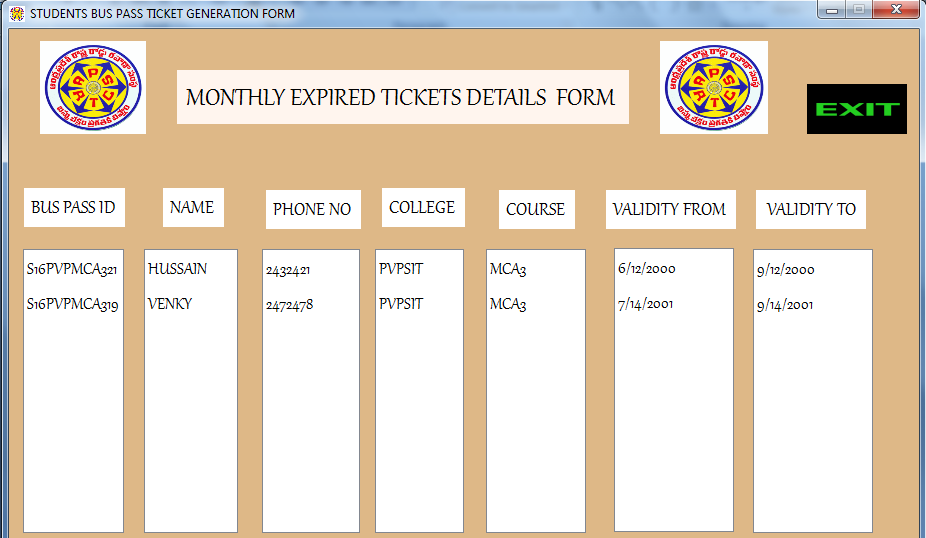
**STUDENT BUS PASS SCREEN**

****

**TICKET GENERATION SCREEN**

****

**EXPIRED TICKETS SCREEN**

****

**6.TESTING**

Software testing is a critical element of software quality assurance and represents the ultimate service of specification design and coding. The increasing visibility of software as a system element and the attended costs associated with the software failure and motivating forces for well planned, through testing. It is not unusual for a software development to spend between 30 and 40 percent of total project effort in testing.

System Testing Strategies for this system integrate test case design techniques into a well-planned series of steps that result in the successful construction of this software. It also provides a road map for the developer, the quality assurance organization and the customer, a roadmap that describes the steps to be conducted as path of testing, when these steps are planned and then undertaken and how much effort, time and resources will be required. The test provisions are follows.

**6.1Testing Objectives:**

The following are the testing objectives....

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has a high probability of finding an as yet undiscovered error.
* A successful test is one that uncovers as a yet undiscovered error.

The above objectives imply a dramatic change in view point. They move counter to the commonly held view that a successful test is one in which no errors are found. Our objective is to design tests that systematically different clauses of errors and do so with minimum amount of time and effort.

If testing is conducted successfully, it will uncover errors in the software. As a secondary benefit, testing demonstrates that software functions appear to be working according to specification and that performance requirements appear to have been met. In addition, data collected as testing is conducted provides a good indication of software.

Testing can't show the absence of defects, it can only show that software errors are present. It is important to keep this stated in mind as testing is being conducted.

**Testing principles:**

Before applying methods to design effective test cases, a software engineer must understand the basic principles that guide software testing.

* All tests should be traceable to customer requirements.
* Tests should be planned ling before testing begins.
* Testing should begin "in the small" and progress towards testing "in the large".

Exhaustive testing is not possible.

**6.2 BLACK BOX TESTING**

**Black-box testing** is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

Here in this application the black box test can be conducted in different ways. Mainly it concentrates on the following:

1. Incorrect date selected
2. Incorrect month selection.
3. Errors in the data structures or external database access.
4. Performance errors
5. Incorrect or missing functions
6. Initialization and termination errors.

Black box testing derives test cases by partitioning the input and the output domain of a program in a manner that provides through test coverage.

Graph based testing methods explore the relationships between and behavior of program objects.

Tests can be conducted at software interface by knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational, at the same time searching for errors in called black box testing, sometimes called as behavioral testing.

Black box testing is not an alternative to white box techniques. Rather it is a complementary approach that is likely to uncover a different class of errors than white box methods. Black box tests are designed to uncover errors in functional requirements without regard to the internal workings of a program. Black box testing techniques focus on the information domain of the software.

Black box testing derives test cases by partitioning the input and output domain of a program in a manner that provides thorough test coverage. Graph based testing methods explore the relationships between and behavior of program objects. Equivalence partitioning divides the input domain into classes of data that are likely to exercise specific software function. Boundary value analysis probes the programs ability to handle data at the limits of acceptability.

**System Testing:**

System tests are designed to validate a fully developed system with a view to assuming that it meets its requirements.

There are three kinds of system testing

1. **Alpha Testing**: Alpha testing refers to the system testing that is carried by the customer within the organization along with the developer. The alpha tests are conducted in controlled manner.
2. **Beta Testing**: Beta testing is the system performed by a selected group of customers, the developer is not present at the site and the user will inform the problems that are encountered during testing. The software developer makes the necessary changes and submits to the customer.
3. **Acceptance Testing**: Acceptance testing is the system testing performed by the customer to whether or not to accept the delivery of the system.
4. **Unit testing** focuses verification effort on the smallest unit of the software design, the module. Using the detailed design description as a guide, important control paths are tested to uncover errors with the boundary of the module for the following modules.All the statements in the module are executed at least once. From this we can ensure that all independent paths through the control structures are exercised.
5. **Integration Testing**:Integration testing is a systematic technique for constructing the program structure and to conduct tests for uncovered errors with interfacing.

In this system, Top-Down integration is performed for the construction of program structures.

1. **Validation Testing:**This testing is performed to ensure that the system functions in a manner that can be reasonably by the users.

Here input data is validated first at interface (authentication), before sending it to the server. In this system, for example user ID and password is checked weather the user is authorized user or not to ensure authentication in each and every module. By this testing we can ensure that all functional requirements are satisfied, and all performance requirements are achieved.

1. **System Testing:**System testing is actually a series of different tests, whose primary purpose is to be fully exercising the computer-based system. In this system, although each test has a different purpose, all the works are verified to ensure that all system elements have been properly integrated and performed allocated functions.Here various system elements like My SQL connectivity and their port numbers and host numbers are calculated.

**6.3 WHITE BOX TESTING**

**White-box testing** (also known as **clear box testing**, **glass box testing**, **transparent box testing**, and **structural testing**) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

1. Guarantee that all independent paths within a module have been exercised at least once.
2. Exercise all logical design on their true or false sides.
3. Execute all loops at their boundaries and within their operational boundaries.
4. Exercise internal data structure to ensure their validity.

Several methods are used for white box testing. The description below describes how this type of testing is applicable to report generation and how the result will form.

1. **Statement Coverage:**In this each statement in a program is executed at least once. This is done by checking the program in debug mode and verifying each statement.
2. **Branch Coverage:** In this each and every branch cases are tested whether the different branch conditions are true or false. It is a strong test criterion over statement coverage testing.
3. **Condition Coverage:** Condition testing is a test case design method that exercises to logical conditions contained in a program module. The purpose of condition testing is to detect not only errors in the condition of a program but also other errors in the program.
4. **Path Coverage:** In this testing all the linearly independent paths in a program are executed once.

**7.CONCLUSION**

This project is helpful to maintain the students bus pass and it reduce the time taken for creating a ID card. It can reduce the Physical presence in the ID card Counter .This project helpful to find the Expired Ticket.

**8.BIBLIOGRAPHY**

**Reference**

1.VISUAL BASIC EXPRESS EDITION TUTORIAL

2.VISUAL BASIC EXPRESS Edtion 5

1.VISUAL BASIC Professtional Edition 3

**Websites:**

[www.Youtube.com/VisualBasic](http://www.Youtube.com/VisualBasic)