



**Dr. M.G.R.
EDUCATIONAL AND RESEARCH INSTITUTE
(Deemed to be University)**
Maduravoyal, Chennai - 600 095. Tamilnadu. India.
(An ISO 9001 : 2015 Certified Institution)



Subject Name: OBJECT ORIENTED SOFTWARE ENGINEERING LAB
Subject code: BCS17L08 / BCS18L08
Programme Name: B.Tech., CSE
Regulation: 2017/2018

STAFF MANUAL

Table of Contents

Ex.No.	Name of the Exercises	Page no.
1	Online Railway reservation system	3
2	Student Mark Analysis system	26
3	Payroll processing application	45
4	Inventory system	71
5	Quiz system	85
6	Automating the Banking process	98
7	Course Registration System	124
8	Library management system	139
9	Passport Automation System	170

EX.NO 1. **ONLINE RAILWAY RESERVATION SYSTEM**

AIM

To develop the Passport Automation System using rational rose tools, visual basic and MS access.

Problem analysis and project planning

Introduction

This document deals with online ticket reservation for railway. This document is designed in such a way that the reader understands it. The use case description and other documents are described in such a way that the system reaches the people easily.

Objectives

The purpose of the document is to know about the availability of seats, railway etc. According to the requirements, passenger reserves his/her tickets.

Scope

This document for online ticket reservation for airline makes the work easy for the passenger to book tickets.

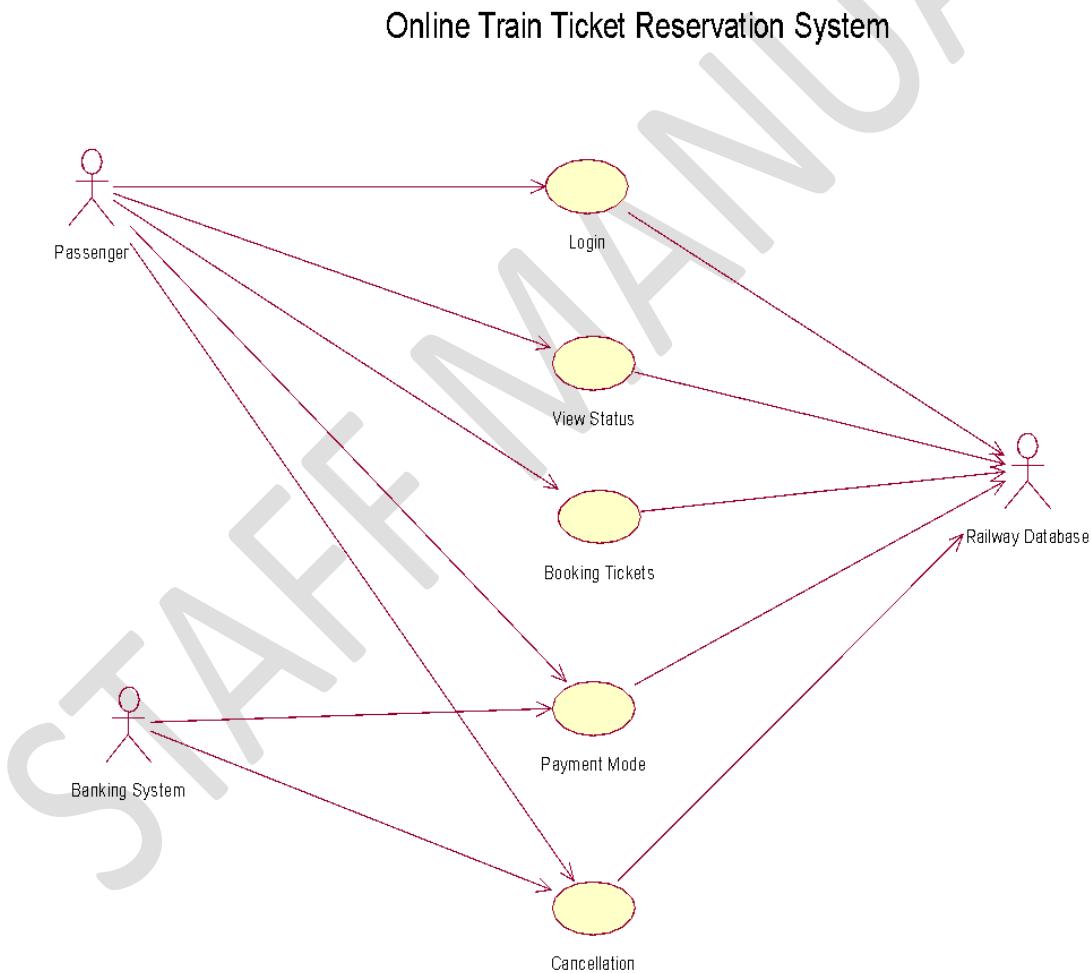
Problem Statement

Computers are an integral part of our day today life. It makes the work easy and faster. Every job is computerized now. So is the ticket reservation, we can book our tickets online. During the reservation of tickets the passenger has to select the origin, date of journey, passport number, etc.

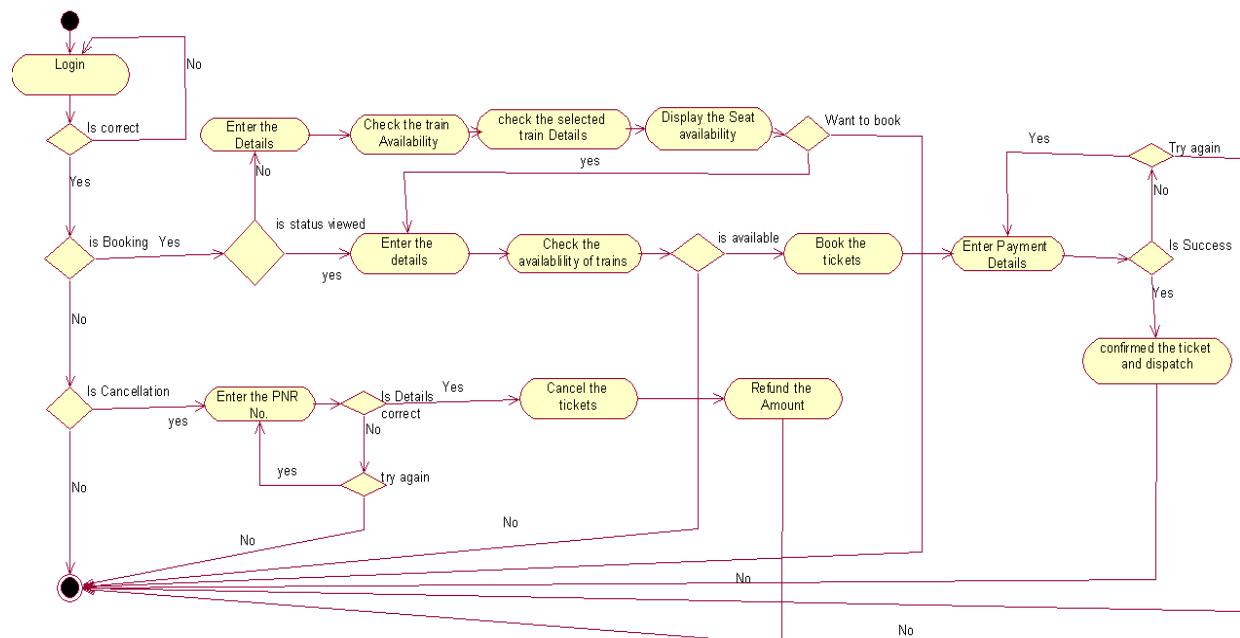
The reservation counter keeps track of the passenger's information. The system will have all the details about the trains and the facilities provided by them. There are various trains provided according to the convenience of the passenger. A database is maintained by the database administrator.

UML DIAGRAMS:

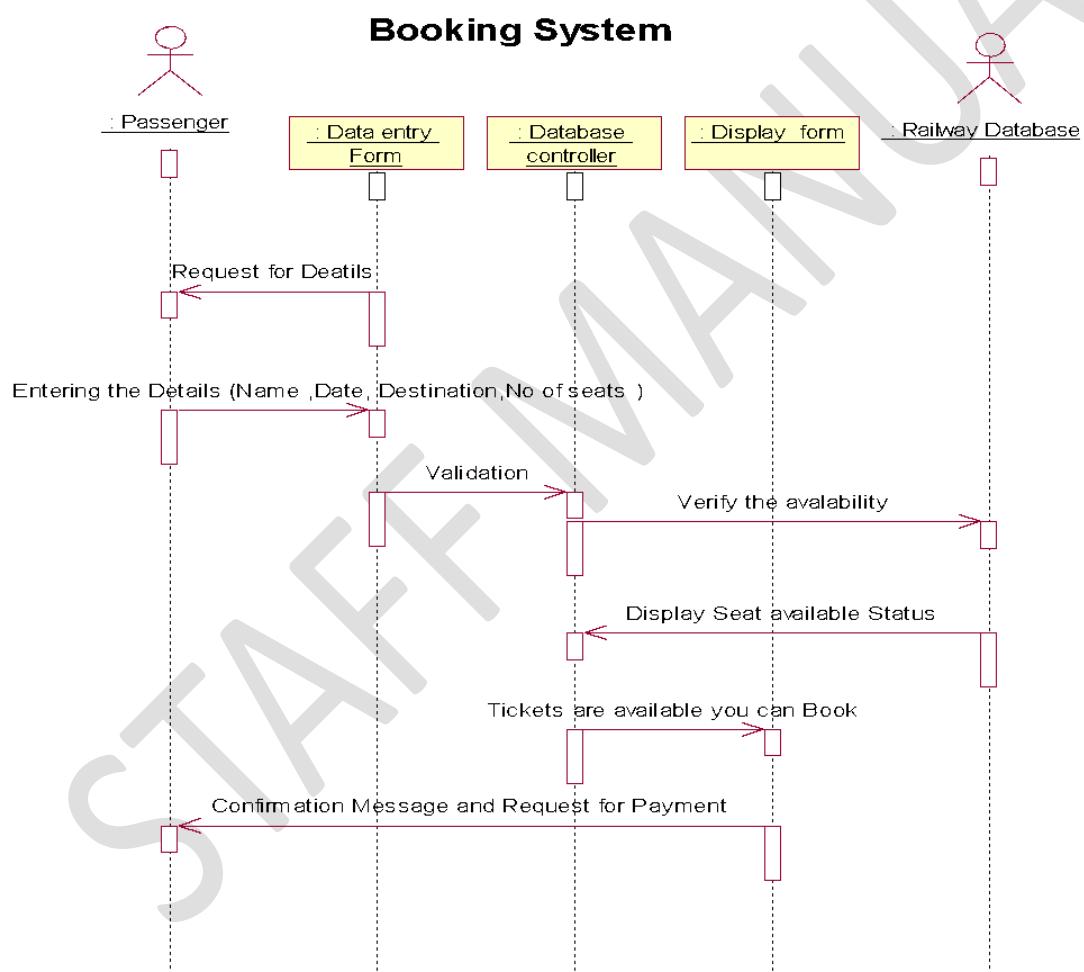
Use Case Diagram

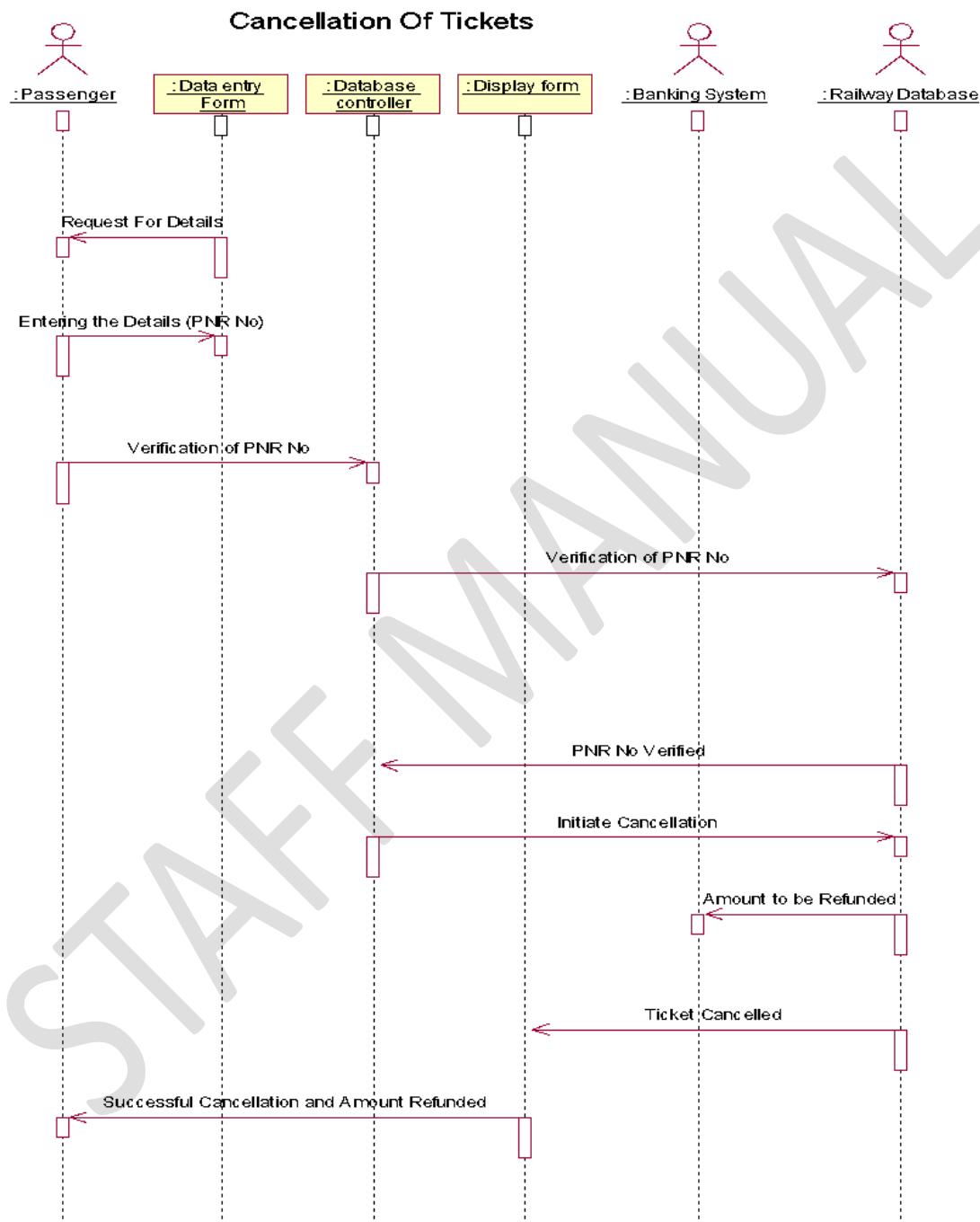


Activity Diagram:

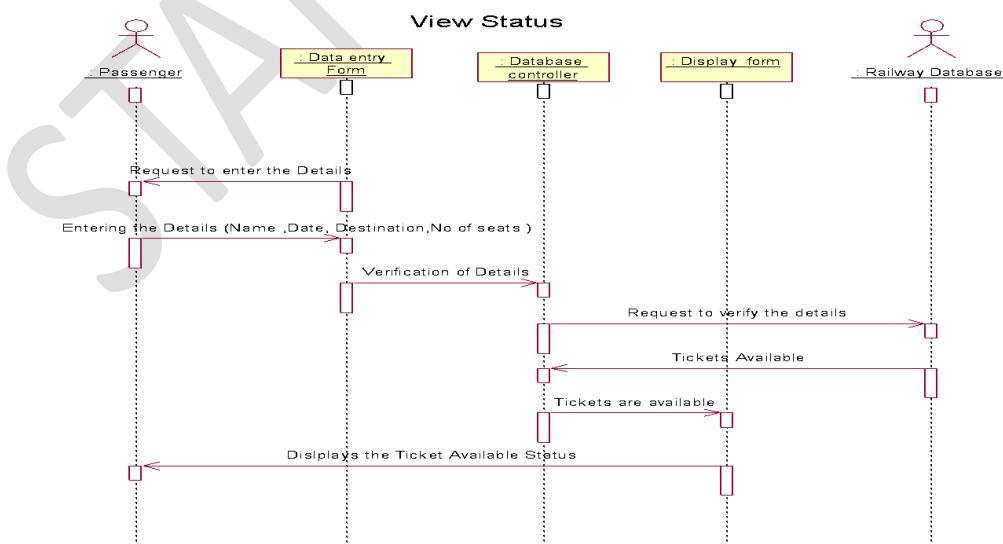
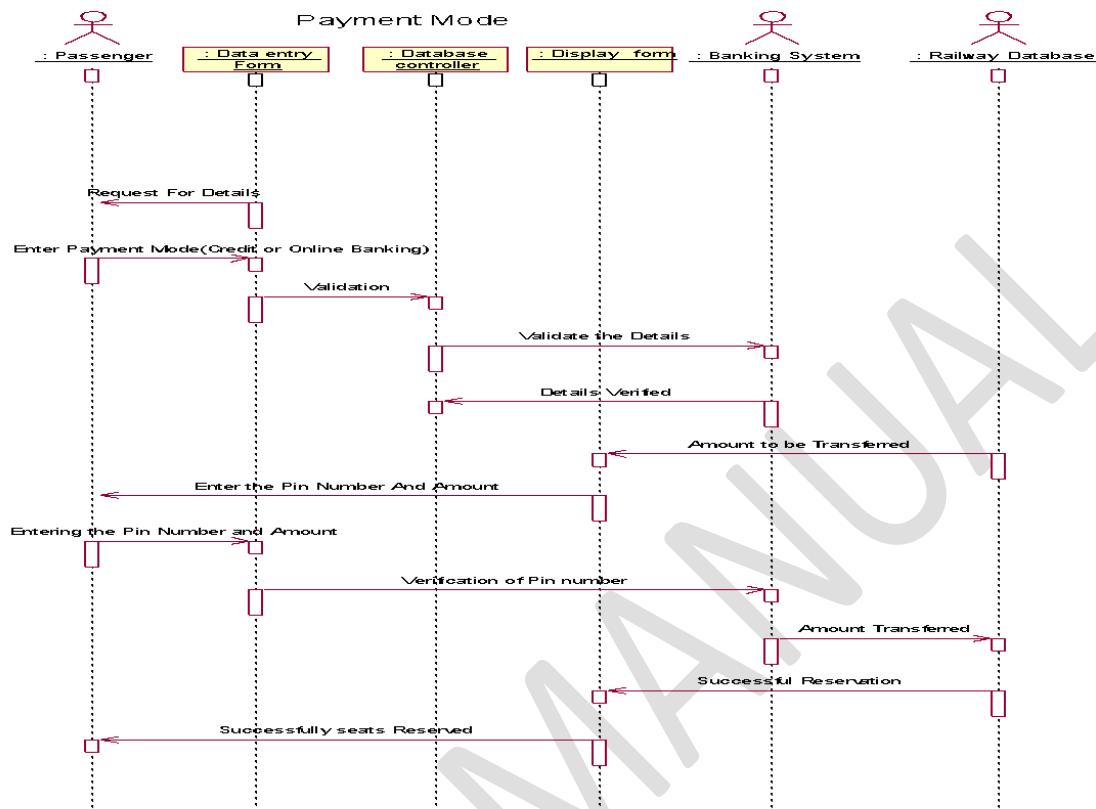


Sequence Diagrams:

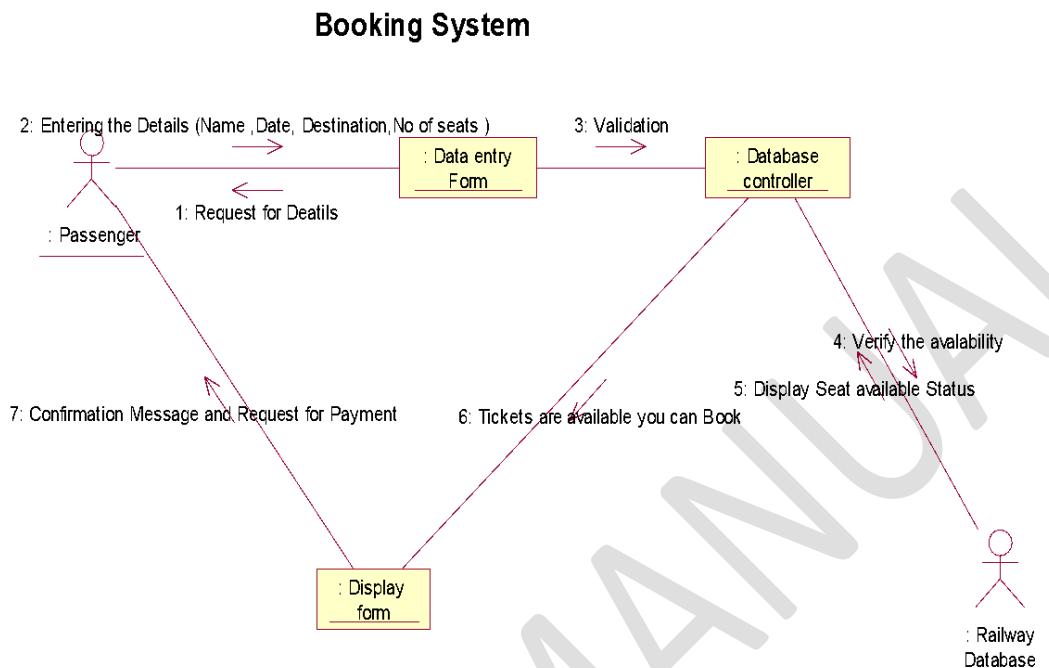




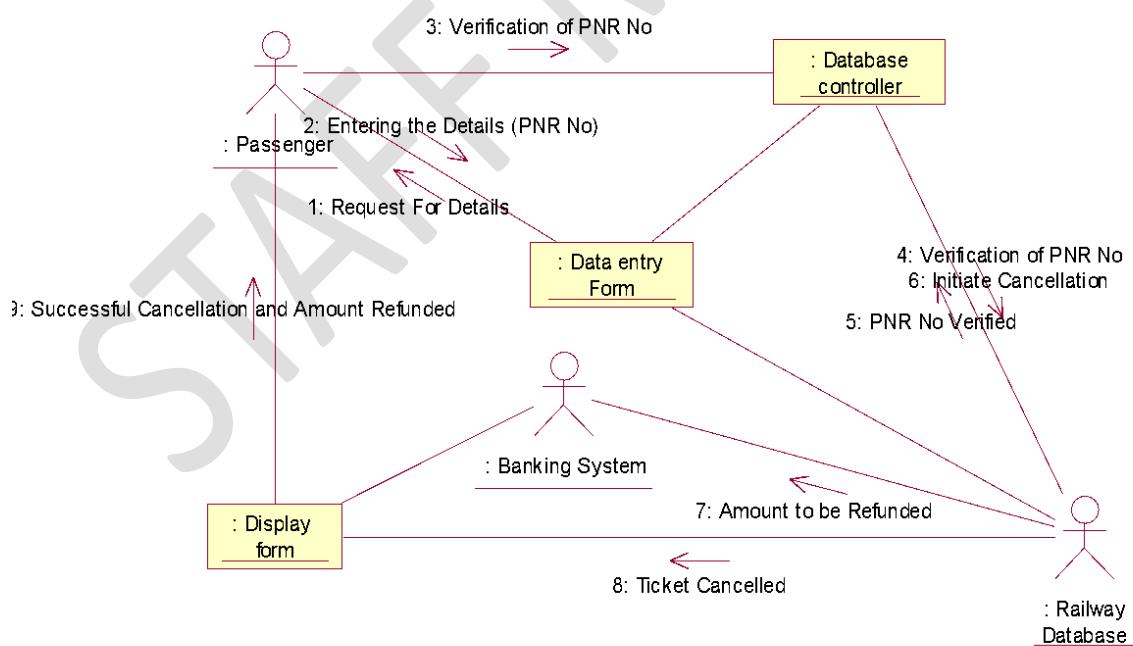
STAFF MANUAL



Collaboration Diagram:

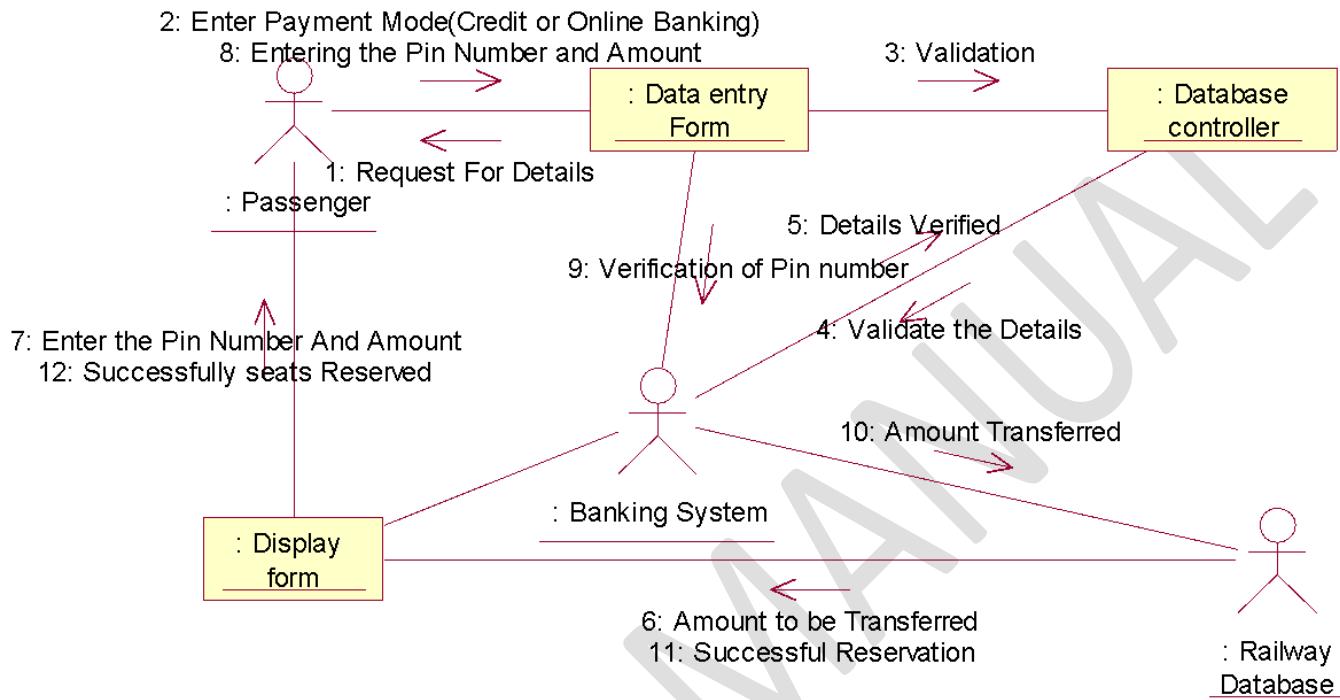


Cancellation of Tickets

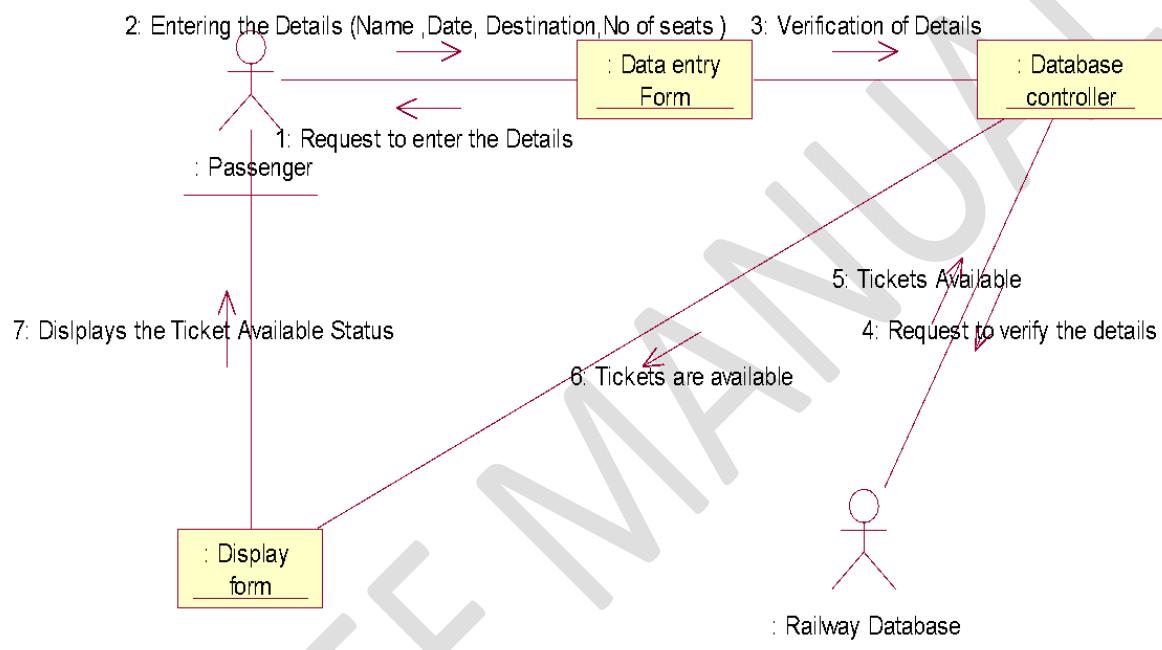


STAFF MANUAL

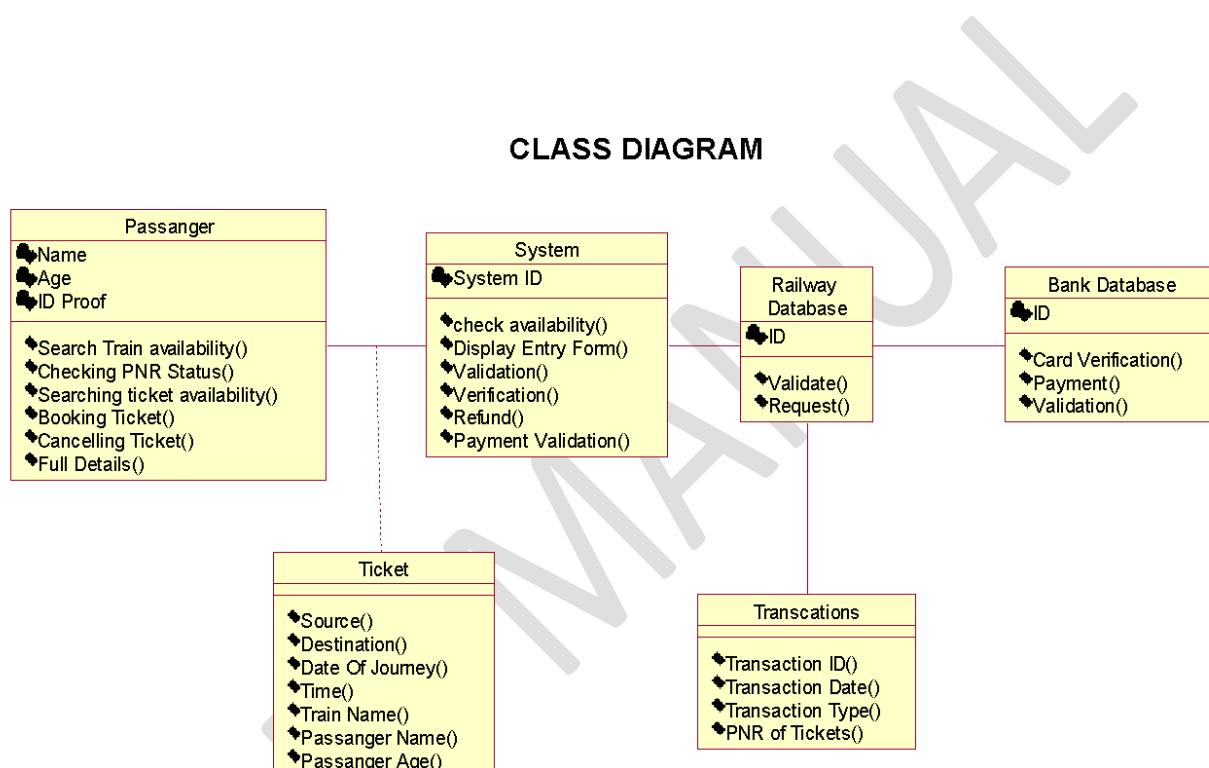
Payment Mode



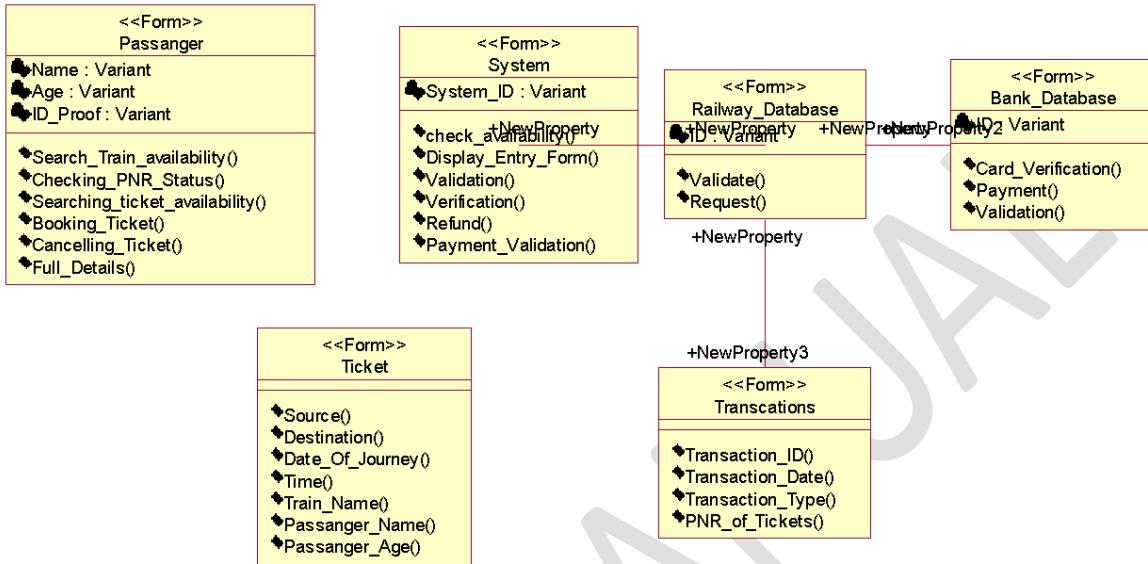
View Status



Class Diagram



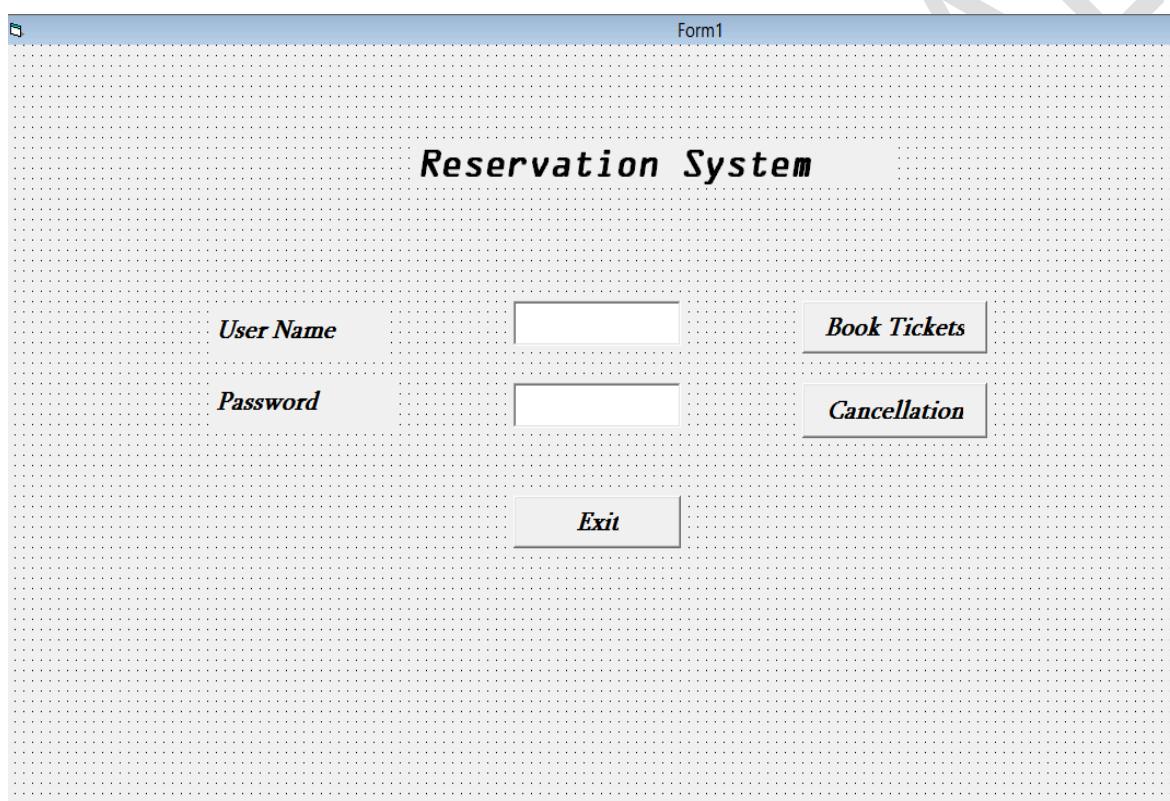
Class Diagram



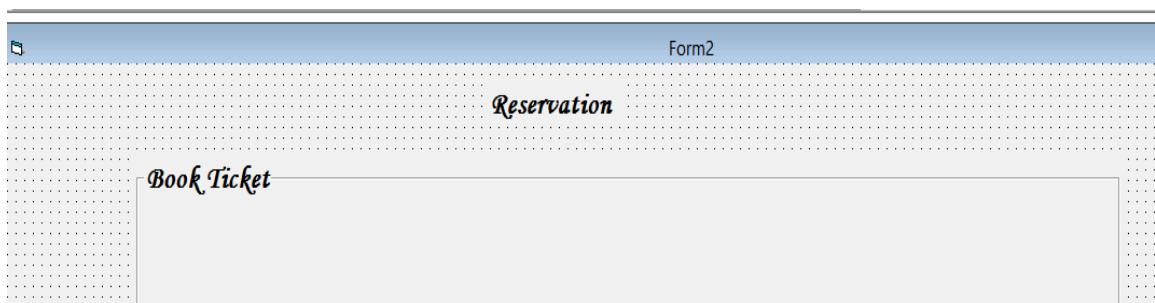
Forward Engineering

Form Design:

Form1:



Form 2:



Form 3:

Form3

Cancellation

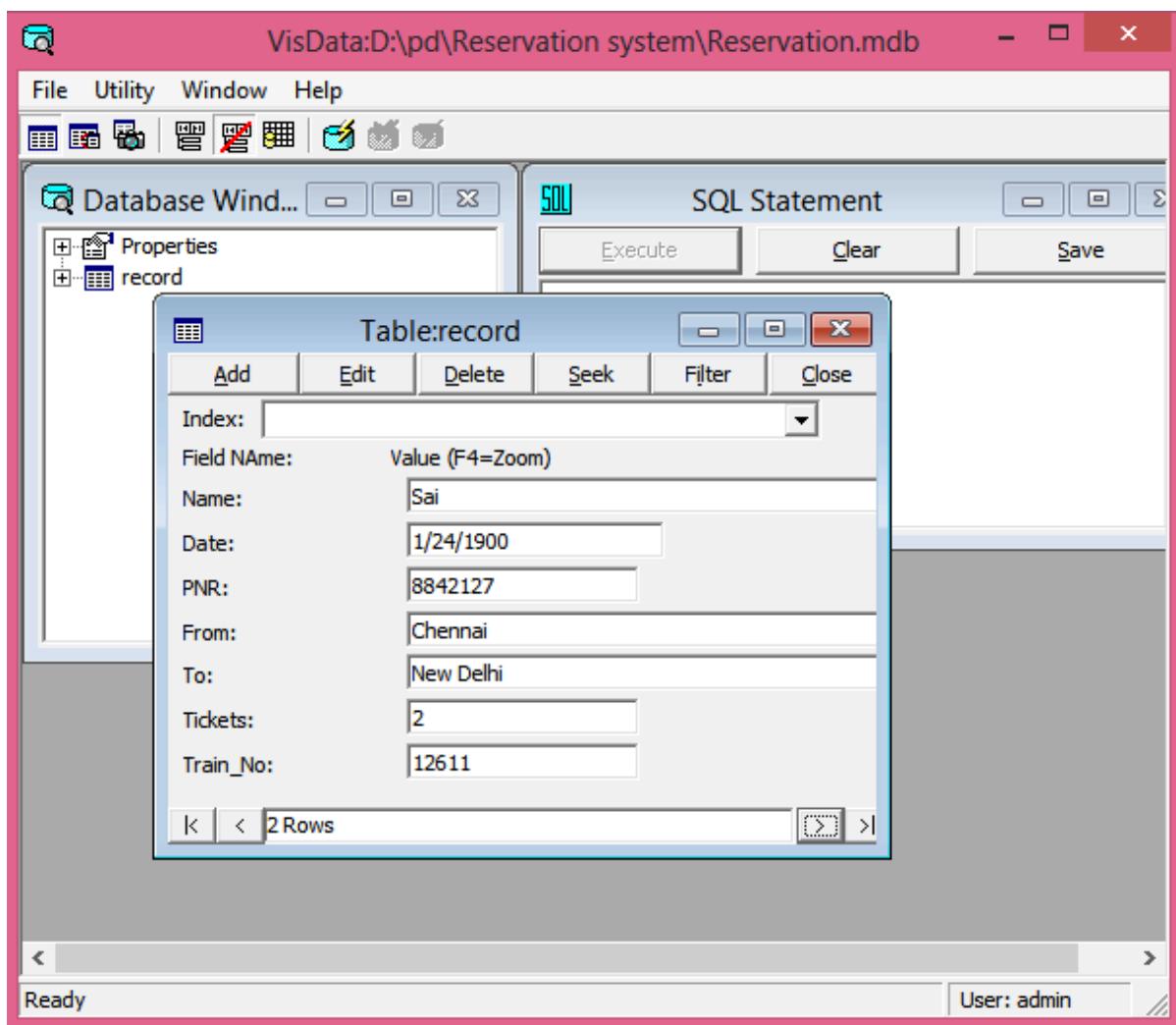
PNR Details

<i>PNR No.</i>	<input type="text"/>	<i>Date</i>	<input type="text"/>
<i>Name</i>	<input type="text"/>	<i>Tickets</i>	<input type="text"/>
<i>Train No.</i>	<input type="text"/>	<i>To</i>	<input type="text"/>
<i>From</i>	<input type="text"/>		

Cancel ***Go Back***



Database:



Coding:

Form 1:

Private Sub BookTickets_Click()

Form2.Show

With Form2

.Text5.Text = ""

.Text2.Text = ""

.Text3.Text = ""

.Text4.Text = ""

.Text6.Text = ""

.Text1.Text = ""

.Text8.Text = ""

End With

Unload Me

End Sub

Private Sub Cancellation_Click()

Form3.Show

Unload Me

End Sub

Private Sub Exit_Click()

End

End Sub

Form 2:

```
Dim db As Database  
Dim RS As Recordset  
Private Sub Command1_Click()  
RS.AddNew  
Frame1.Visible = True  
Frame2.Visible = False  
Frame3.Visible = True  
End Sub
```

```
Private Sub Command2_Click()  
Form1.Show  
Unload Me  
End Sub
```

```
Private Sub Command3_Click()  
Frame1.Visible = False  
Frame2.Visible = True  
Frame3.Visible = False
```

```
Dim intResult As Currency  
  
intResult = Int((10000000 * Rnd) + 1)  
Text8.Text = intResult  
MsgBox "Ticket Booked" '  
On Error Resume Next  
RS.Fields(0) = Trim(Text1)
```

```
RS.Fields("Date") = Val(Text5)  
RS.Fields("Train_No") = Val(Text2)  
RS.Fields("PNR") = Val(Text8)  
RS.Fields("From") = Trim(Text3)  
RS.Fields("To") = Trim(Text6)  
RS.Fields("Tickets") = Val(Text4)  
RS.Update
```

End Sub

```
Private Sub Command4_Click()  
Frame1.Visible = True  
Frame2.Visible = False  
Frame3.Visible = False  
End Sub
```

```
Private Sub Exit_Click()  
End  
End Sub
```

```
Private Sub Form_Load()  
Frame1.Visible = True  
Frame2.Visible = False  
Frame3.Visible = False
```

```
Set db = OpenDatabase("D:\pd\Reservation system\Reservation.mdb")  
Set RS = db.OpenRecordset("record")
```

Text5.Text = RS!Date

Text2.Text = RS!Train_No

Text3.Text = RS!From

Text4.Text = RS!Tickets

Text1.Text = RS!Name

Text6.Text = RS!To

Text8.Text = RS!PNR

End Sub

Private Sub Main_Click()

Form1.Show

Unload Me

End Sub

Form 3:

Private Sub Command1_Click()

With Form2.Data1

.Recordset.MoveFirst

Do Until .Recordset.EOF

If .Recordset.Fields("PNR") = Text1.Text Then

Frame2.Visible = False

Frame1.Visible = True

Text2.Text = .Recordset.Fields("PNR")

Text3.Text = .Recordset.Fields("Name")

Text4.Text = .Recordset.Fields("Train_No")

Text5.Text = .Recordset.Fields("From")

Text6.Text = .Recordset.Fields("Date")

Text7.Text = .Recordset.Fields("Tickets")

Text8.Text = .Recordset.Fields("To")

Exit Sub

Else

.Recordset.MoveNext

End If

Loop

MsgBox "Not Found", vbCritical, "Failed"

Text1.Text = ""

Text1.SetFocus

End With

End Sub

Private Sub Command2_Click()

Form2.Data1.Recordset.Delete

MsgBox "Ticket Cancelled "

Form1.Show

Unload Me

End Sub

Private Sub Command3_Click()

Form1.Show

Unload Me

End Sub

```
Private Sub Command4_Click()  
Frame2.Visible = False  
Frame1.Visible = True
```

End Sub

```
Private Sub Form_Load()  
Frame2.Visible = True  
Frame1.Visible = False
```

End Sub

Output:

MANUAL

Form1

Reservation System

User Name	<i>Saireddy</i>	Book Tickets
Password	*****	<i>Cancellation</i>
Exit		

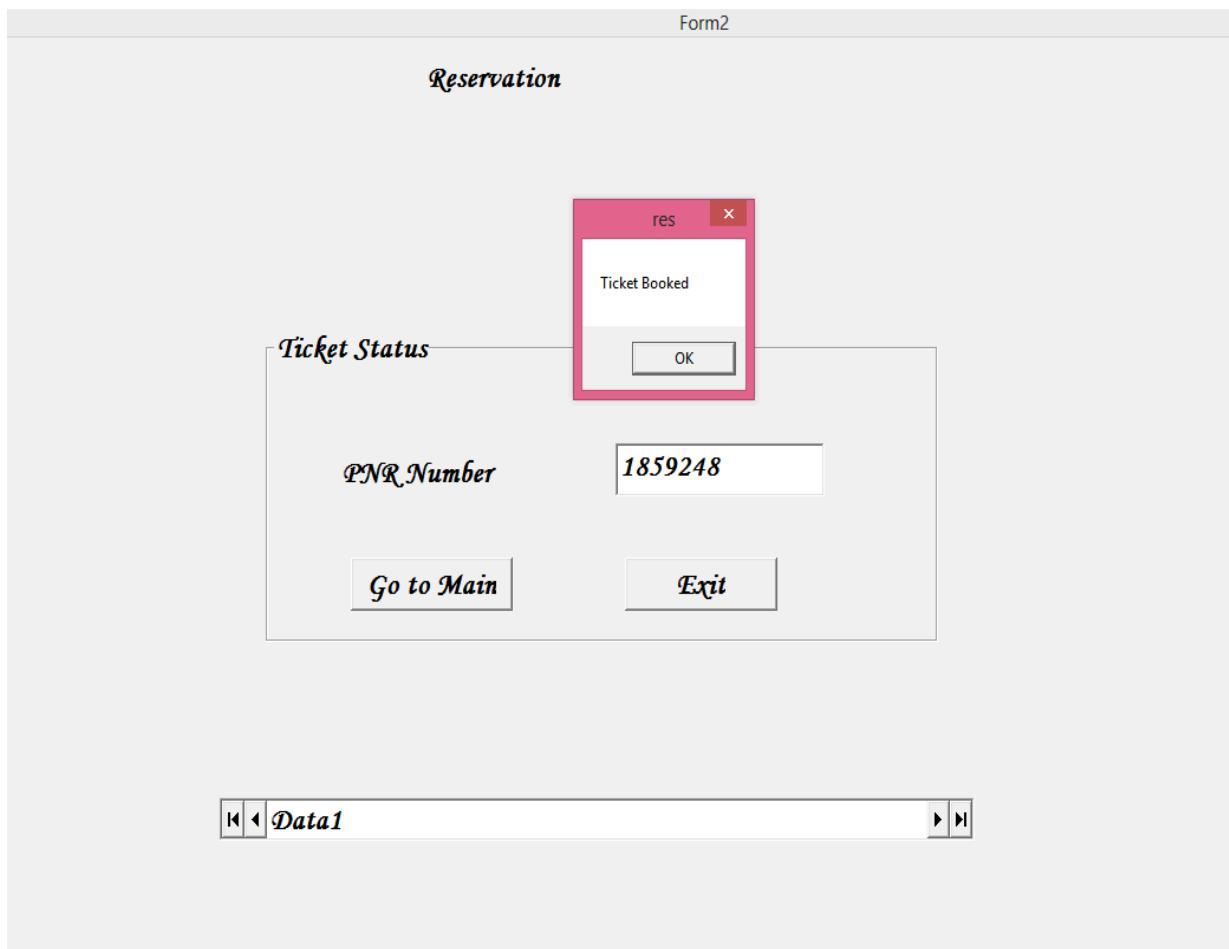
Form2

Reservation

Book Ticket

Name	<input type="text" value="Sai"/>	Date	<input type="text" value="26/5/2013"/>
Train No	<input type="text" value="12611"/>		
From	<input type="text" value="Chennai"/>	To	<input type="text" value="New Delhi"/>
No. Of Tickets	<input type="text" value="1"/>		

STAFF M...



STAFF

Form3

Cancellation

PNR No.

[**Go For Cancellation**](#) [**Go Back**](#)

Form3

Cancellation

PNR Details

PNR No.	1859248	Date	1/8/1900
Name	Ratnesh	Tickets	2
Train No.	12423	To	Meerut
From	Banglore		

Cancel **Go Back**

res

Ticket Cancelled

OK

Result:

Thus the Online Ticket Reservation System project is analyzed and designed using Rational Rose and Implemented in VB.

EX.NO2.

STUDENT MARK ANALYSIS SYSTEM

AIM

To develop the Student Mark Analysis System, visual basic and MS access.

Problem analysis and project planning

Introduction

Student mark analyzing system has been designed to carry out the mark analysis process in an educational institution. The results of respective departments can be efficiently computed without much manual involvement.

Objectives

The purpose of this project is to define the requirements of mark analysis system. This system reduces manual work to a great extent. The mark analysis is carried out by the system in an efficient manner.

Scope

This system is very essential for every educational institution as it reduces man power. This system can be used for all kinds of educational institutions to evaluate and analyze the marks and generate reports of specified criteria.

Problem Statement

For analyzing the marks obtained by students in an educational institution. We are tasked to build up student mark analyzing system. This is done to replace the manual entering and processing of marks which are error prone and tedious. This system also maintains information about students.

The system will have a Windows based desktop interface to allow the faculty to enter marks obtained by the students, update them and generate various reports.

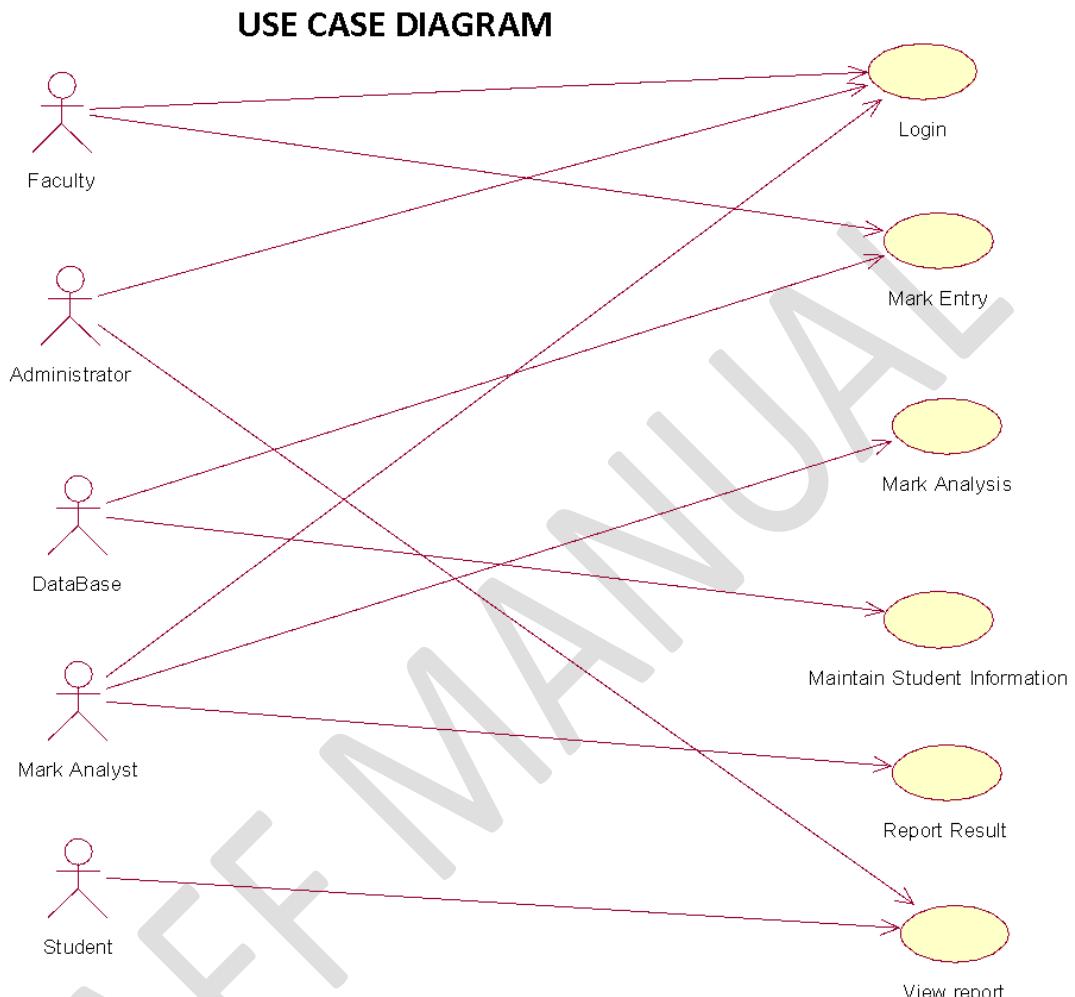
For security reasons, the administrator and faculty only can update the marks and other information. First the user needs to login to the system for accessing it. The system will retain information on all the students and the institution. The system analyses the marks and generates the result reports. The marks and information about the students are stored in a database and the system works with the database.

The faculty can enter the marks and student information through a visual environment. The updated details are stored in the database. The system generates the overall result by analyzing the marks.

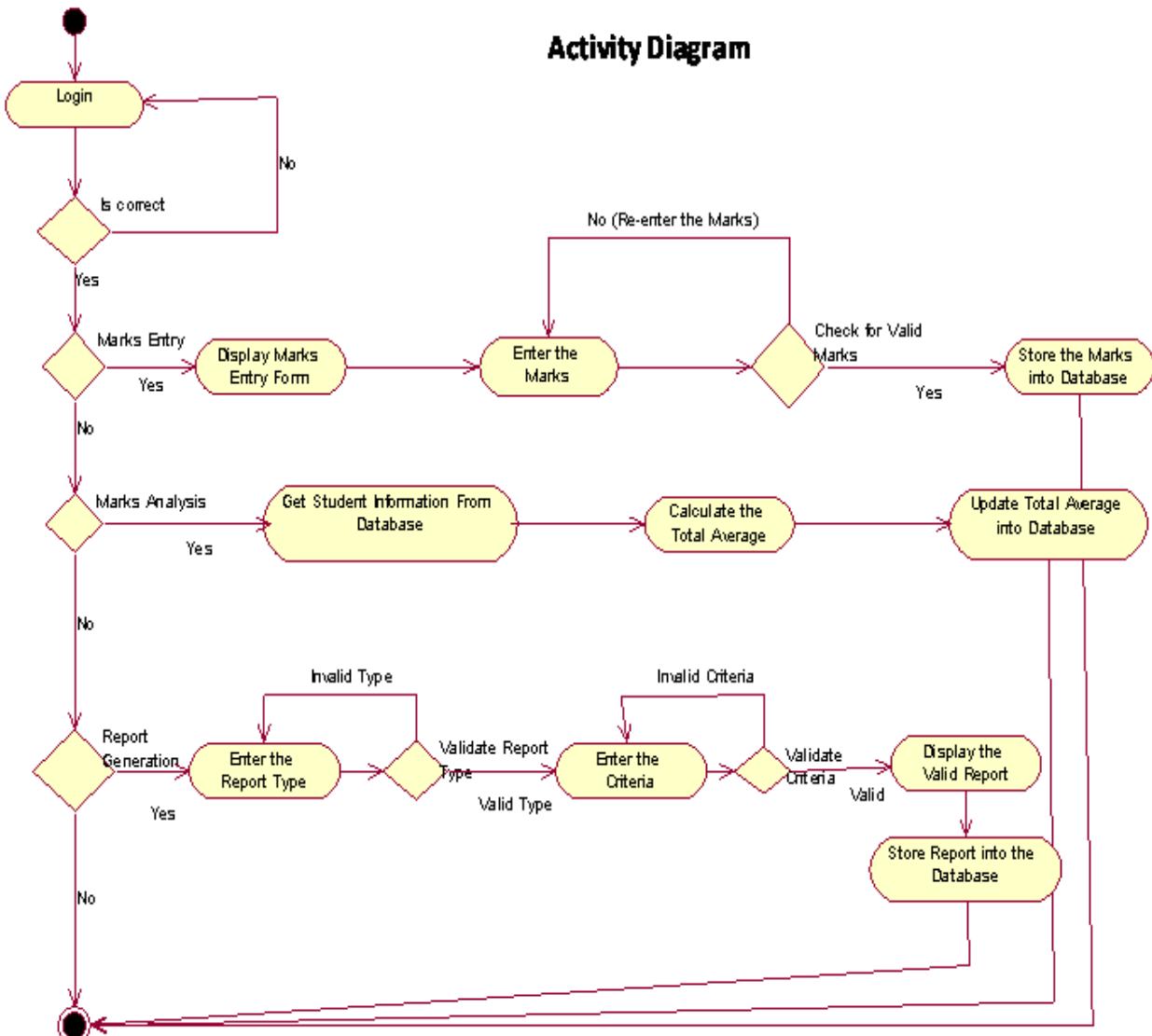
Mark analyzer monitors this process. One of the most important features of the system is creating reports based on the given criteria. The user can create the following reports:

Overall Class, Department result, Individual student result, Toppers list, Arrears list and Improvement rate for the academic year report has to be generated by entering the register number of the student. These reports can also be viewed by the management and placement officers. The administrator is responsible for adding, deleting student details from the system and updating the marks to the system with the external queries. So, the system design will generate reports automatically and there will be no need for manual intervention.

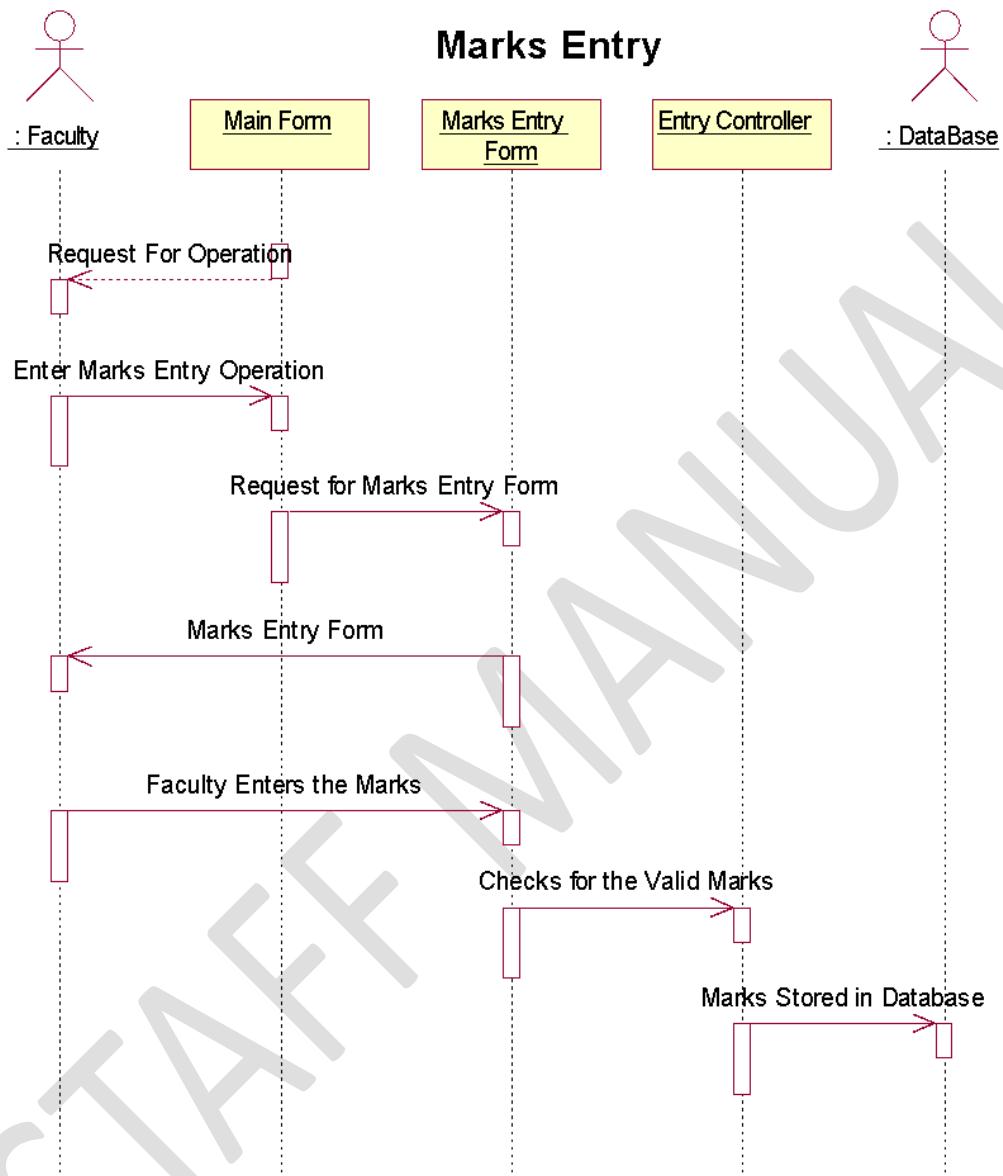
UML Diagrams:

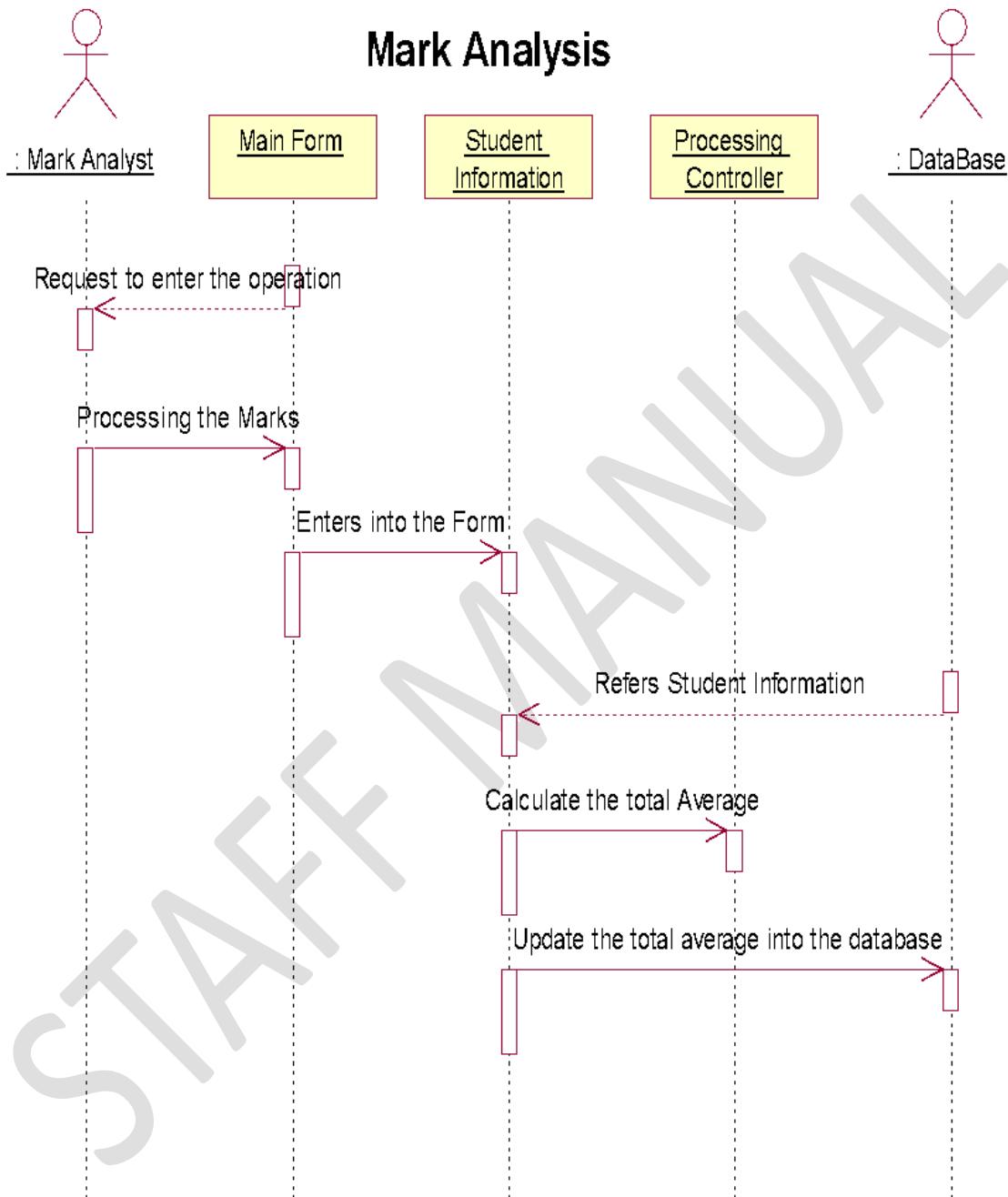


Activity Diagram:

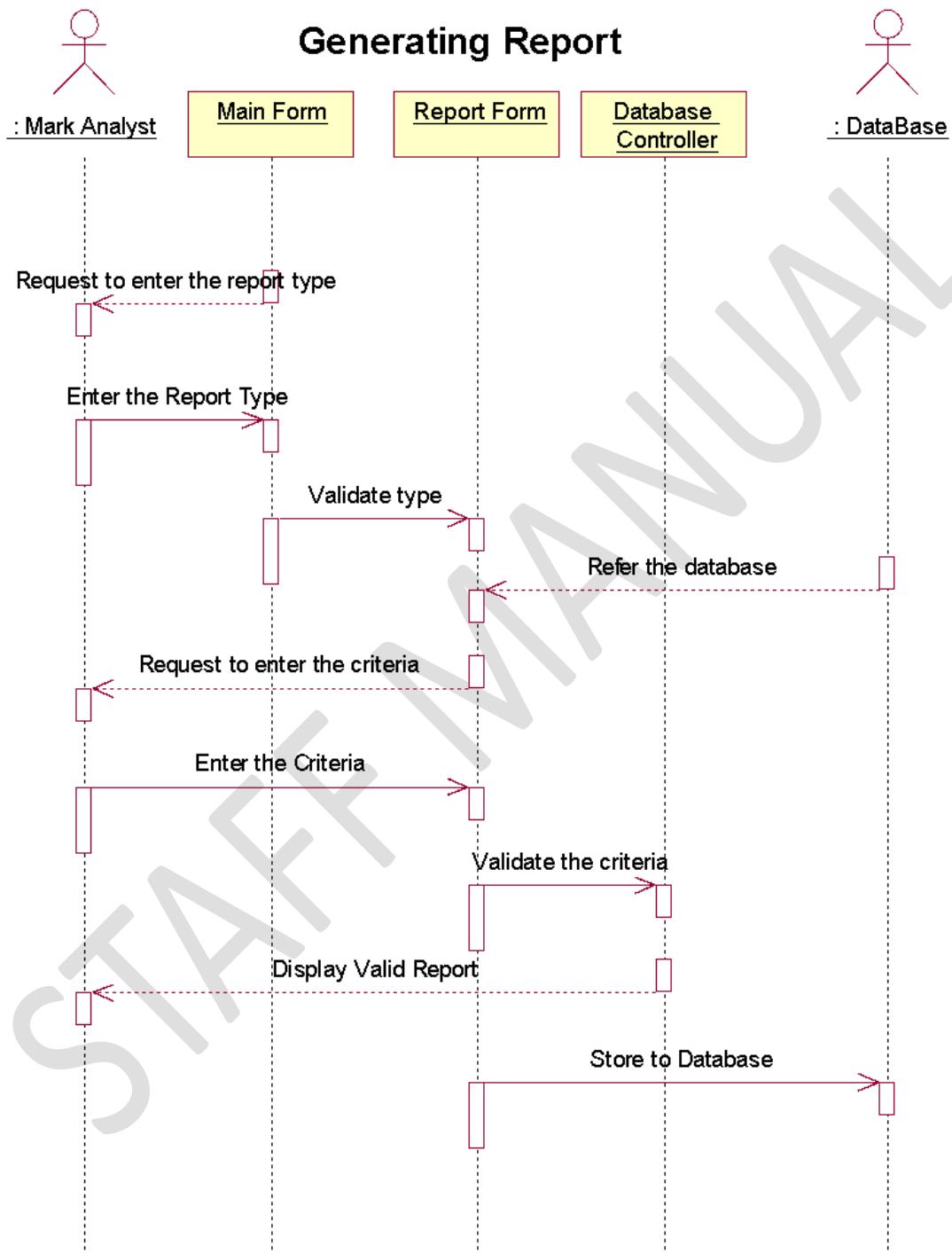


Sequence Diagrams:

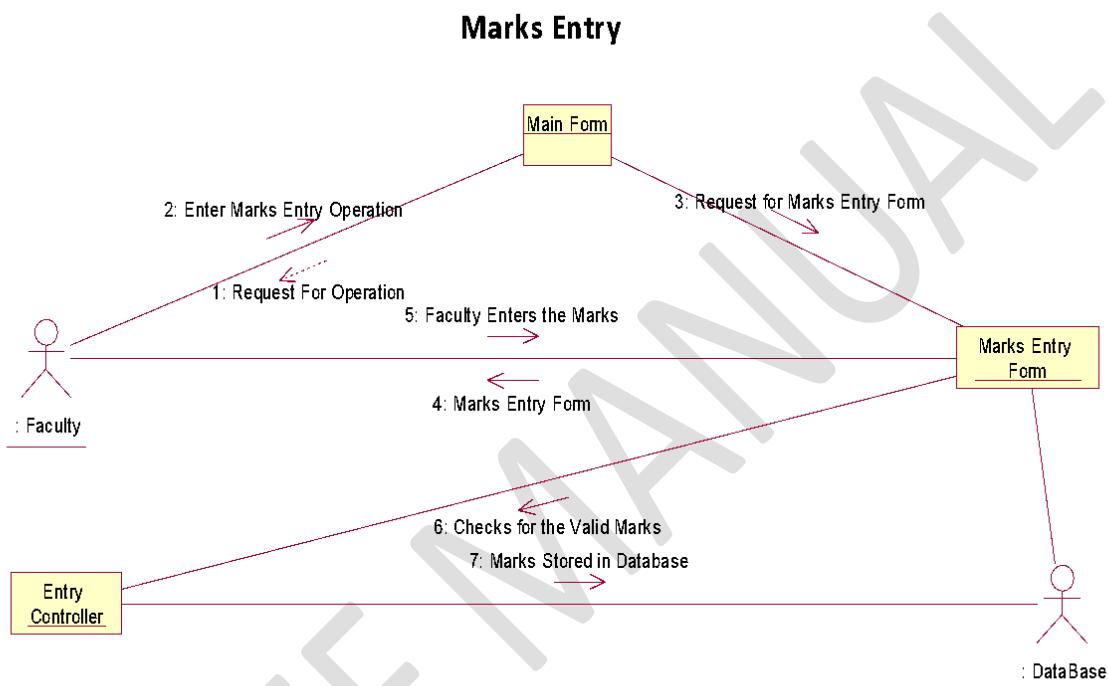




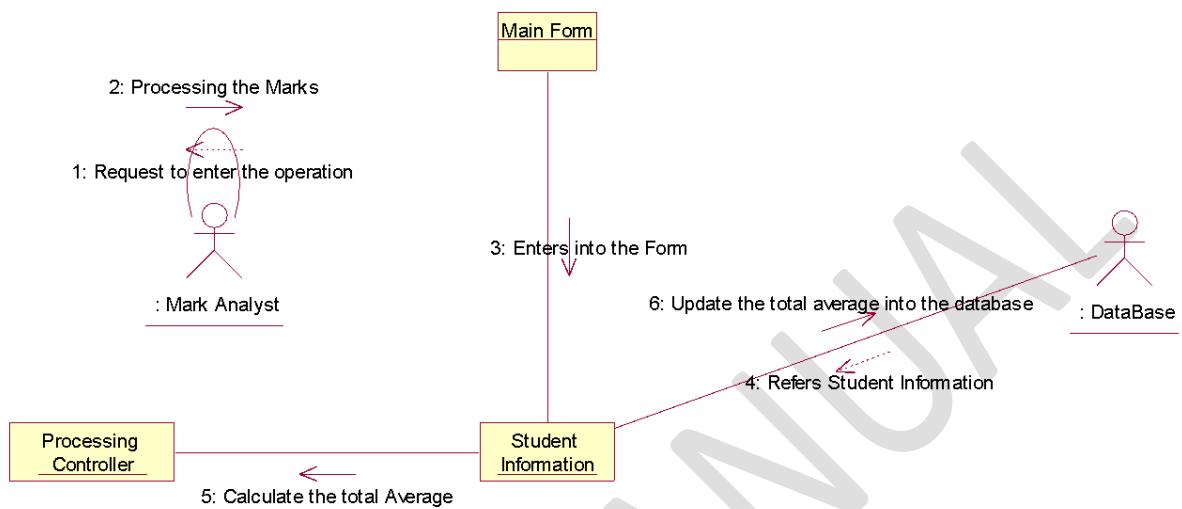
STAFF MANUAL



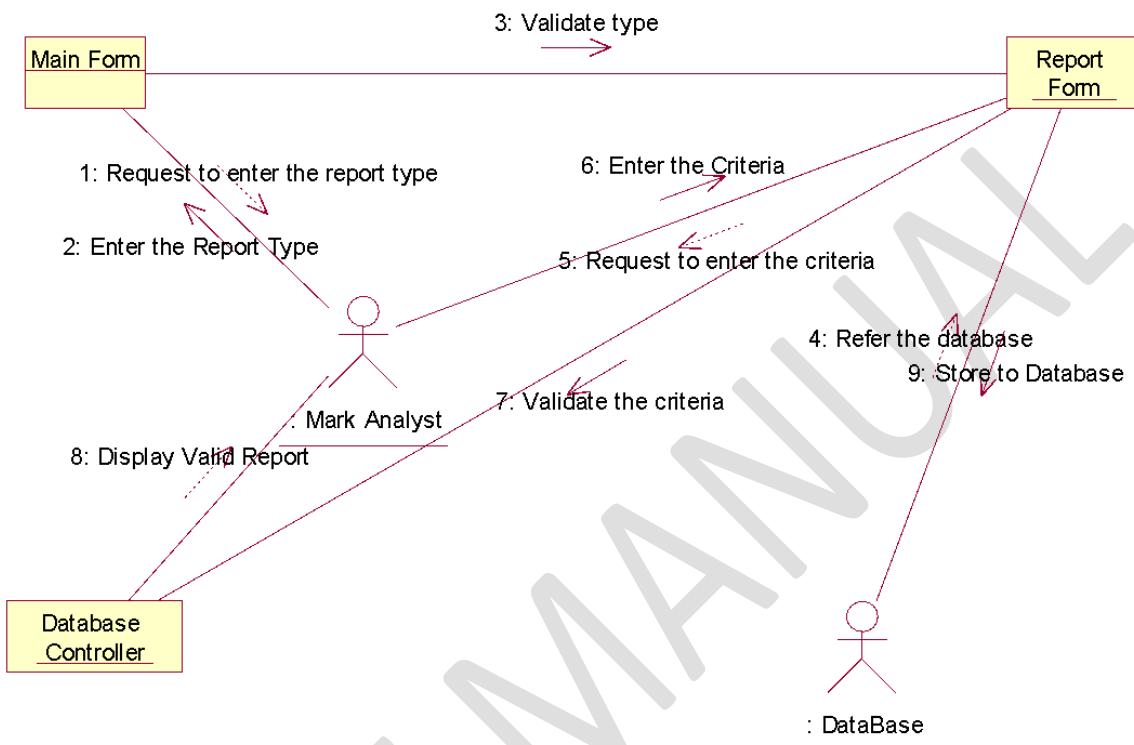
Collaborations Diagrams:



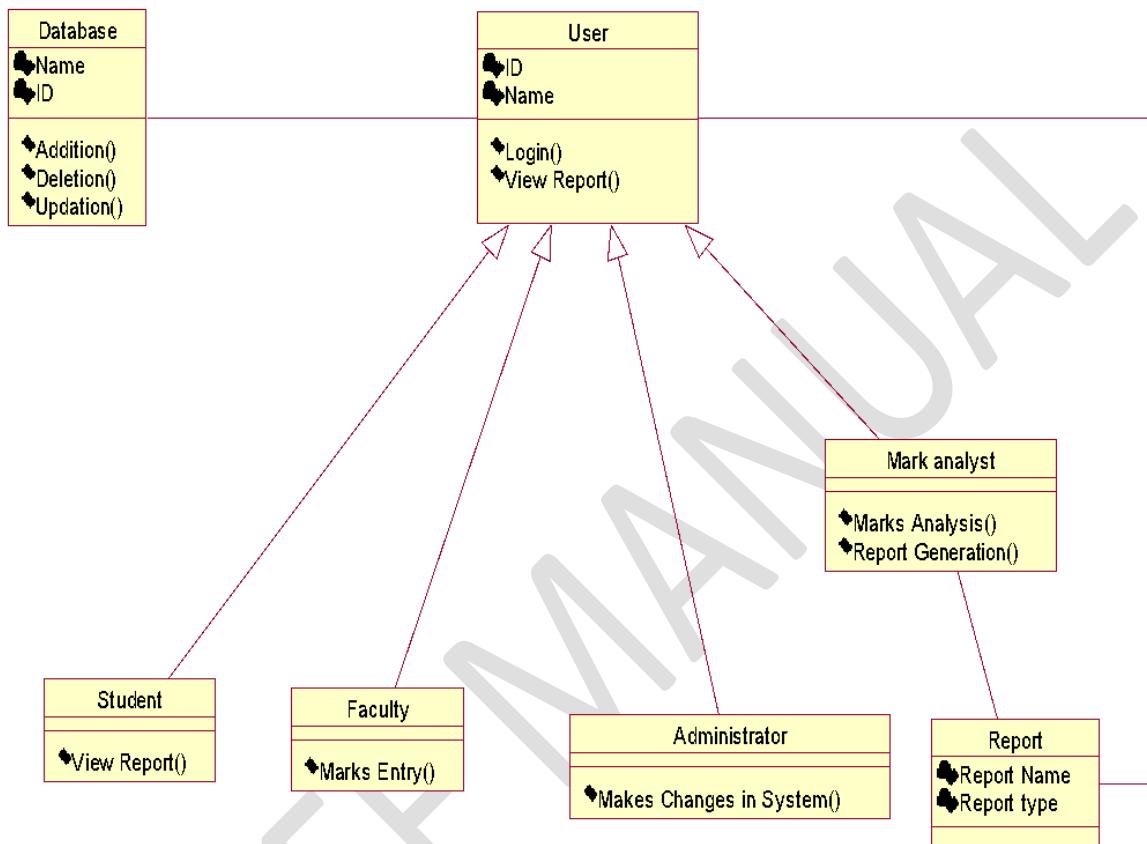
Marks Analysis



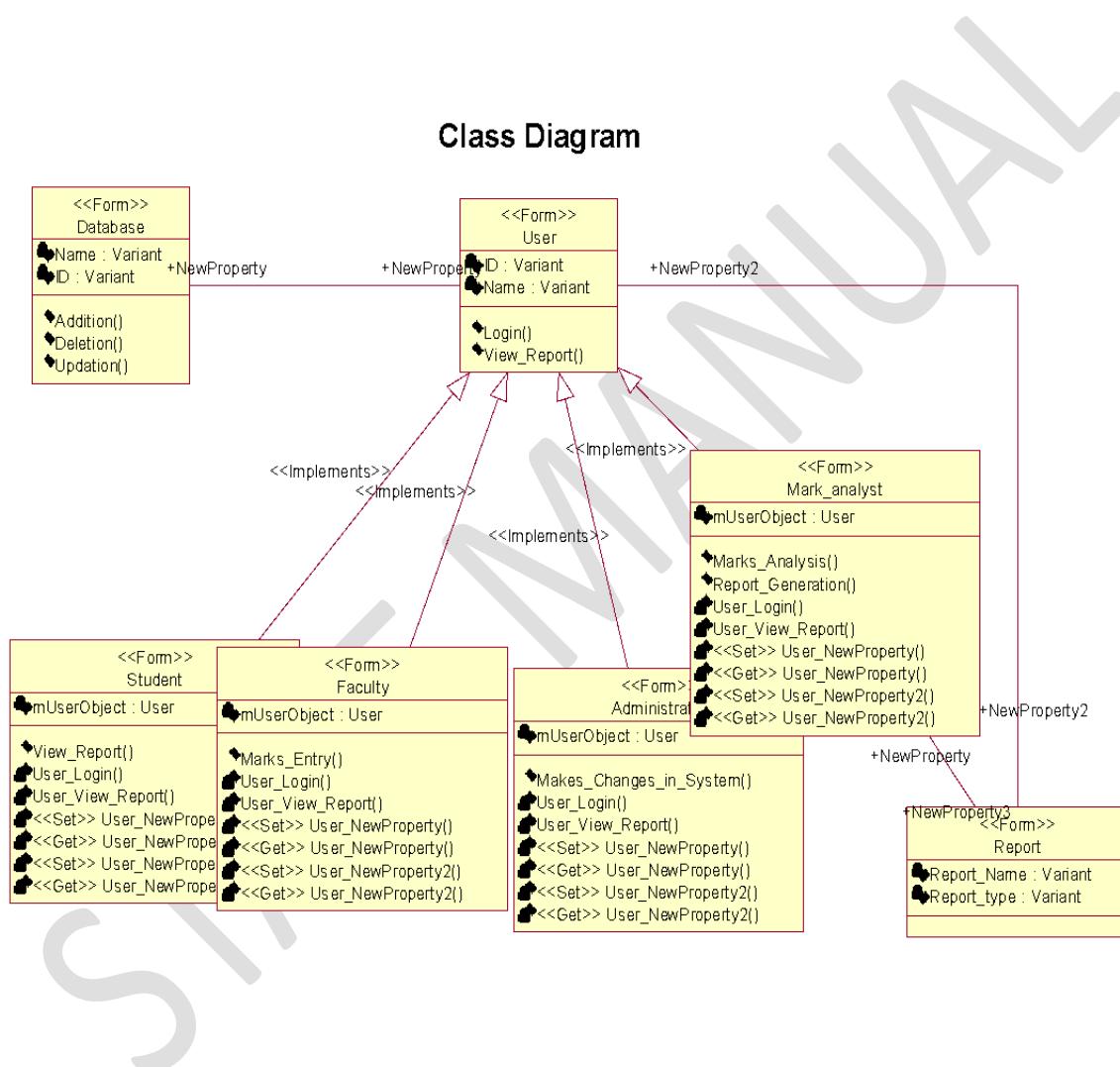
Generating Report



Class Diagram



FORWARD ENGINEERING (Class Diagram)



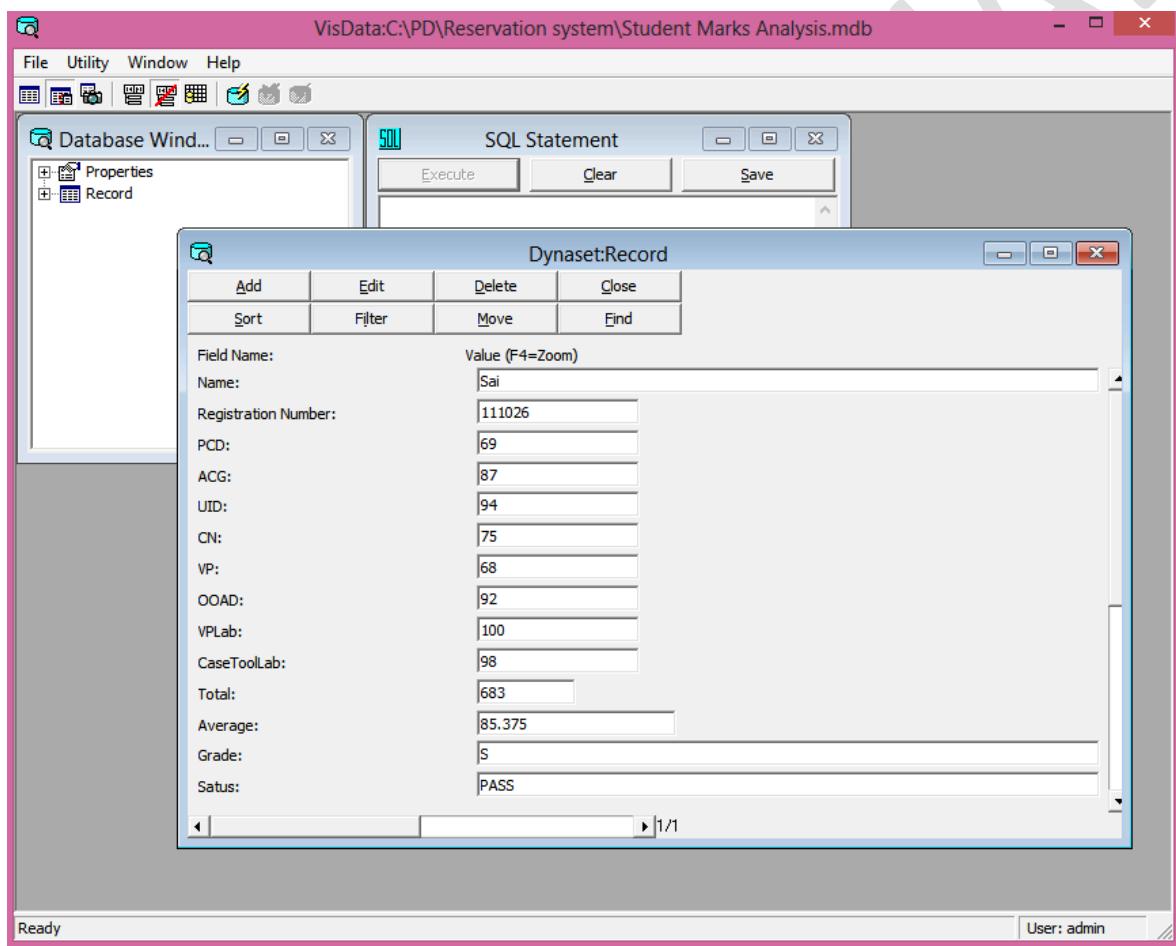
Form Design:

Form1

Name	<input type="text"/>	CN	<input type="text"/>	Total	<input type="text"/>
Registration No.	<input type="text"/>	VP	<input type="text"/>	Average	<input type="text"/>
PCD	<input type="text"/>	OOD	<input type="text"/>	Grade	<input type="text"/>
ACG	<input type="text"/>	VP Lab	<input type="text"/>	Status	<input type="text"/>
UID	<input type="text"/>	Case Tool Lab	<input type="text"/>		

Calculate Clear Save Exit

DATABASE:



Coding

```
Private Sub Command1_Click()
```

```
End Sub
```

```
Private Sub calculate_Click()
```

```
Text11.Text = Val(Text3.Text) + Val(Text4.Text) + Val(Text5.Text) +  
Val(Text6.Text) + Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text)  
+ Val(Text10.Text)
```

```
Text12.Text = Val(Text11.Text) / 8
```

```
If (Val(Text3.Text) >= 50) And (Val(Text4.Text) >= 50) And  
(Val(Text5.Text) >= 50) And (Val(Text6.Text) >= 50) And  
(Val(Text7.Text) >= 50) And (Val(Text8.Text) >= 50) And  
(Val(Text9.Text) >= 50) And (Val(Text10.Text) >= 50) Then
```

```
Text14.Text = "Pass"
```

```
Else
```

```
Text14.Text = "Fail"
```

```
End If
```

```
If Val(Text12.Text) >= 90 Then
```

```
Text13.Text = "H"
```

```
ElseIf Val(Text12.Text) >= 80 Then
```

```
Text13.Text = "S"
```

```
ElseIf Val(Text12.Text) >= 70 Then
```

```
Text13.Text = "A"
```

```
ElseIf Val(Text12.Text) >= 60 Then
```

```
Text13.Text = "B"
```

```
ElseIf Val(Text12.Text) >= 50 Then
```

```
Text13.Text = "C"  
ElseIf Val(Text12.Text) < 50 Then  
Text13.Text = "F"  
End If  
End Sub
```

```
Private Sub Command3_Click()  
End Sub
```

```
Private Sub Clear_Click()  
Data1.Recordset.AddNew  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
Text4.Text = ""  
Text5.Text = ""  
Text6.Text = ""  
Text7.Text = ""  
Text8.Text = ""  
Text9.Text = ""  
Text10.Text = ""  
Text11.Text = ""  
Text12.Text = ""  
Text13.Text = ""  
Text14.Text = ""  
End Sub
```

Private Sub Exit_Click()

End

End Sub

Private Sub save_Click()

Data1.Recordset.Update

Data1.Recordset.MoveLast

MsgBox "Record Saved"

End Sub

OUTPUT

The screenshot shows a Windows application window titled "Form1". Inside the window, there is a grid of input fields for student grades. The columns are labeled "Name", "CN", "Total", "Registration No.", "VP", "Average", "PCD", "OOAD", "Grade", "ACG", "VP Lab", "Status", and "UID". The "Status" column contains a message box titled "Project1" with the text "Record Saved" and an "OK" button. At the bottom of the window are four buttons: "Calculate", "Clear", "Save", and "Exit".

Name	CN	Total	Registration No.	VP	Average	PCD	OOAD	Grade	ACG	VP Lab	Status	UID
Sai	75	683	111026	68	85.375	69	92	S	87	100	PASS	94

Result:

Thus the Student Mark Analysis project is analyzed and designed using Rational Rose and Implemented in VB.

Ex.No.3

Payroll System

AIM

To develop the Payroll System using rational rose tools, visual basic and MS access.

Introduction

This document deals with online Payroll Processing System. This document is designed in such a way that the reader understands it. The use case description and other documents are described in such a way that the system reaches the people easily.

Objectives

To develop a user friendly online pay roll processing system which makes the work easy and faster .

Scope :

This project describes about the payroll system, there are two types of Users, normal users and Administrator. Users can log on the directory by a unique User ID that is provided to them and for every

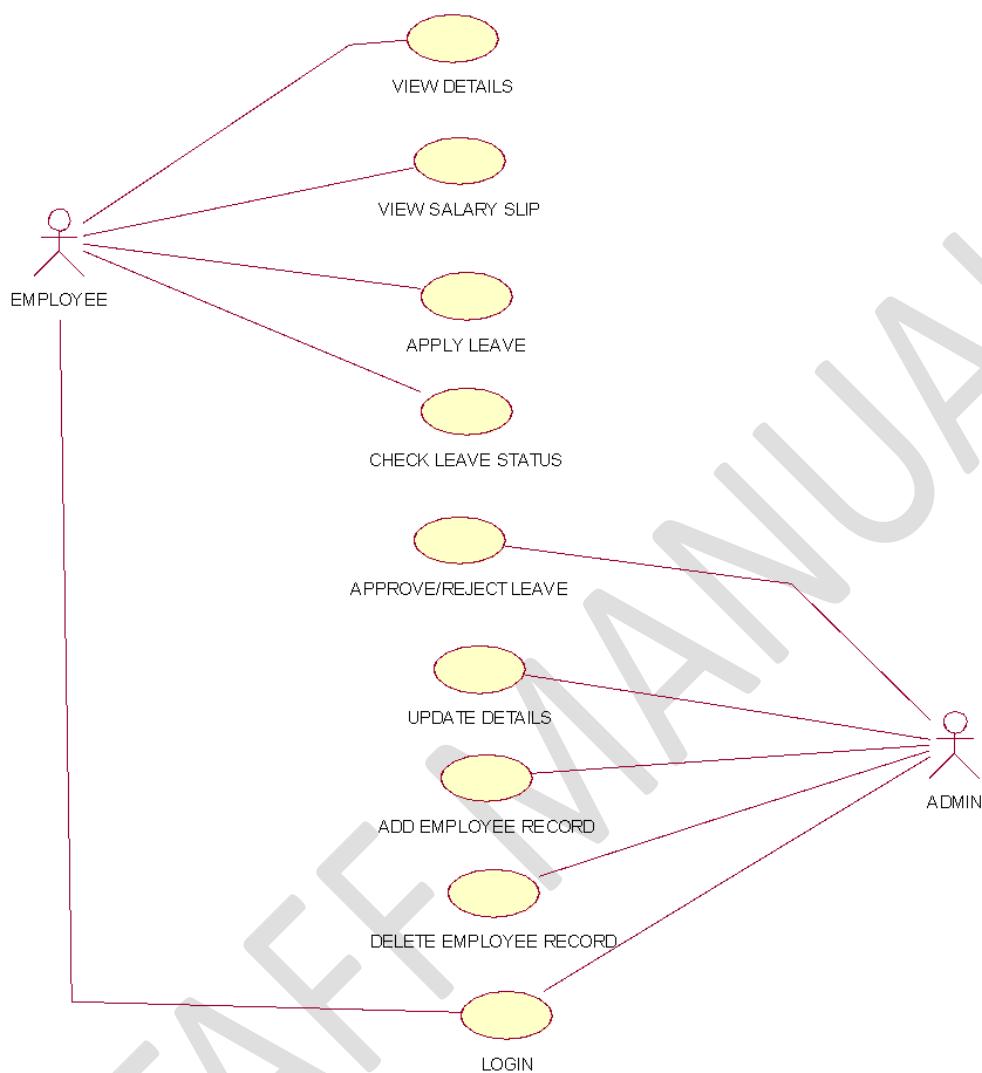
ID check there is a password. Administrator does authorization, he is the manager of that directory, and he can Search employee details with names and their employee id. This would help users to Create or find employee details easily.

Problem Statement:

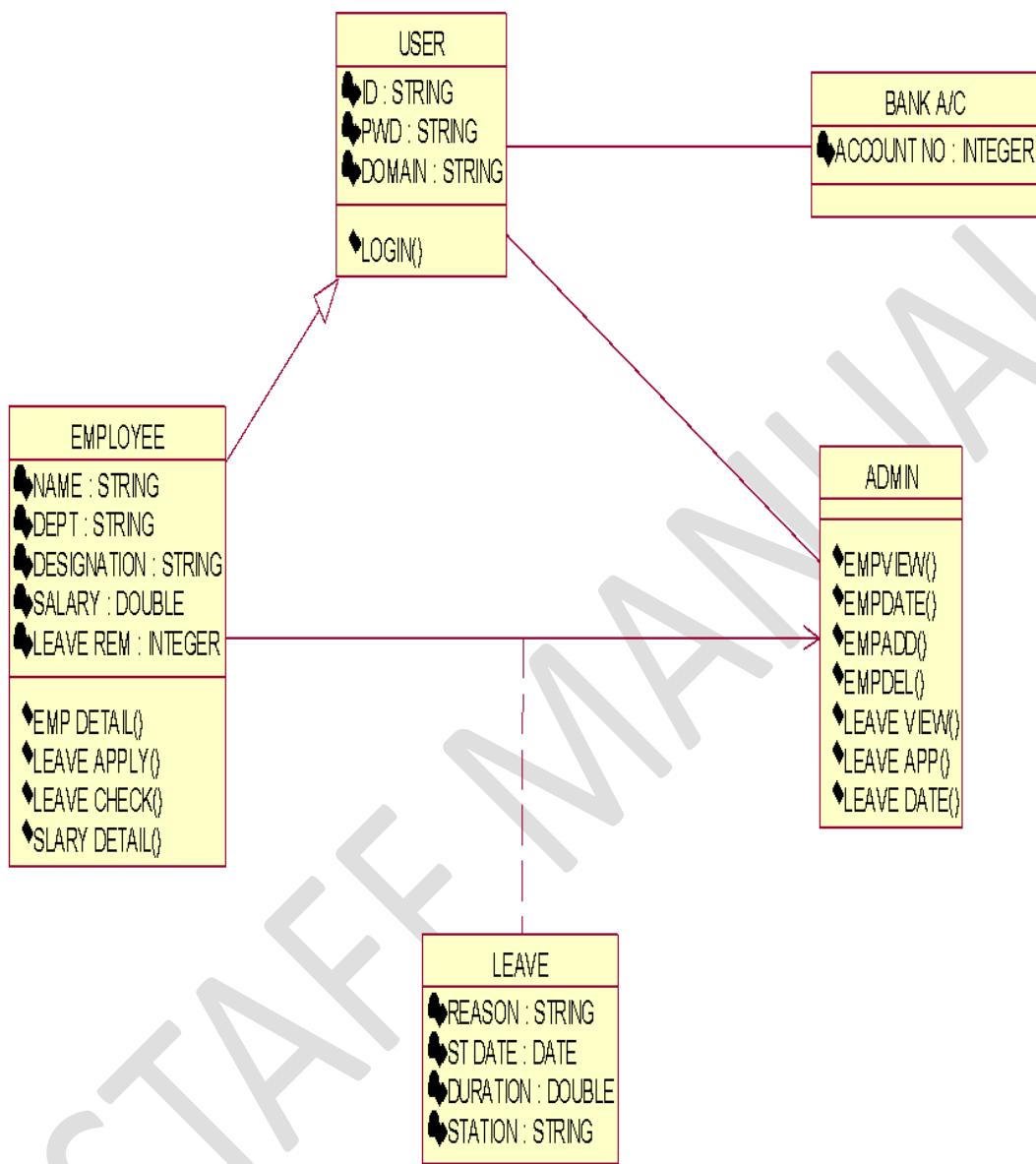
Designing a payroll system for users and administrator through Authentication Process, where only administrator can alter the data in Oracle database, Checking ID's and password of authorized users.

UML DIAGRAMS:-

USE CASE DIAGRAM



CLASS DIAGRAM

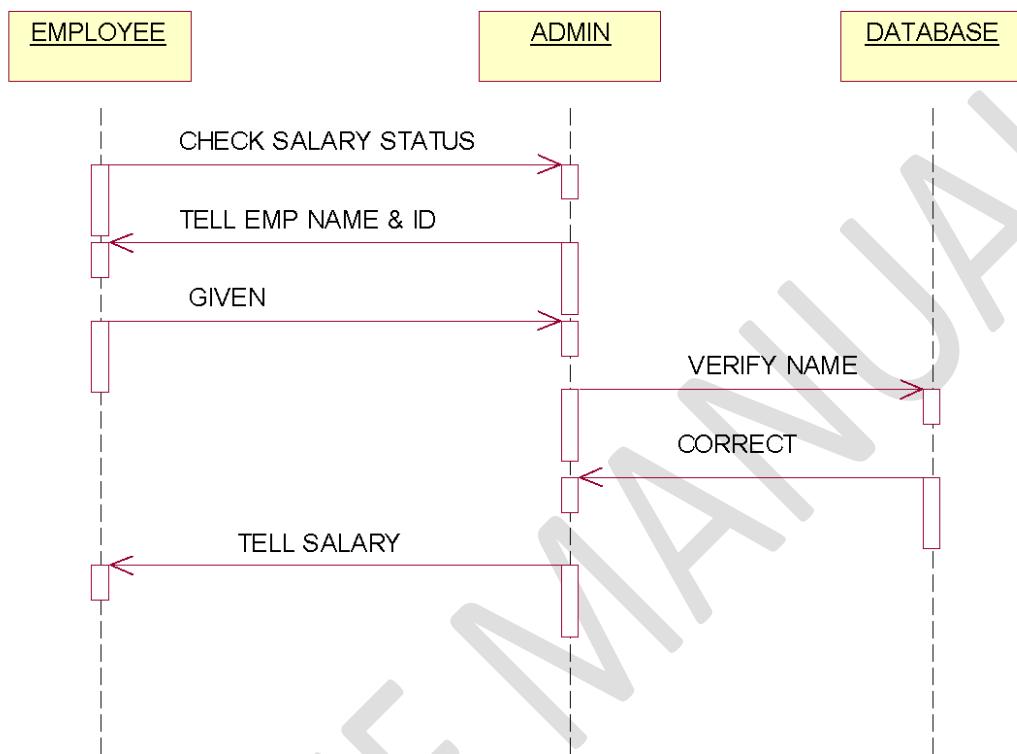


ACTIVITY DIAGRAM

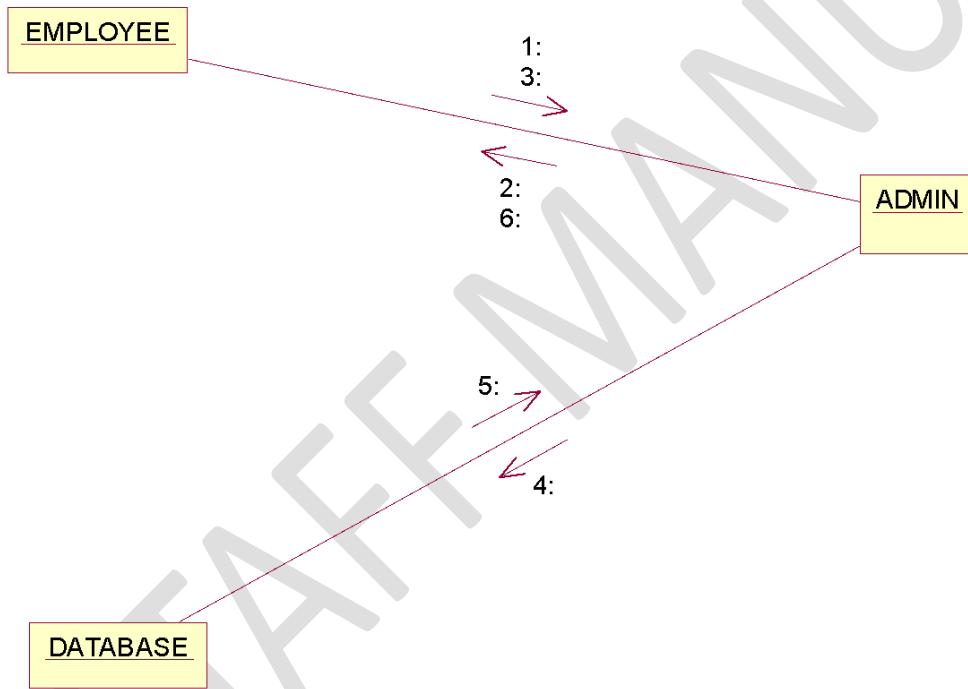


STK

SEQUENCE DIAGRAM: for view salary slip

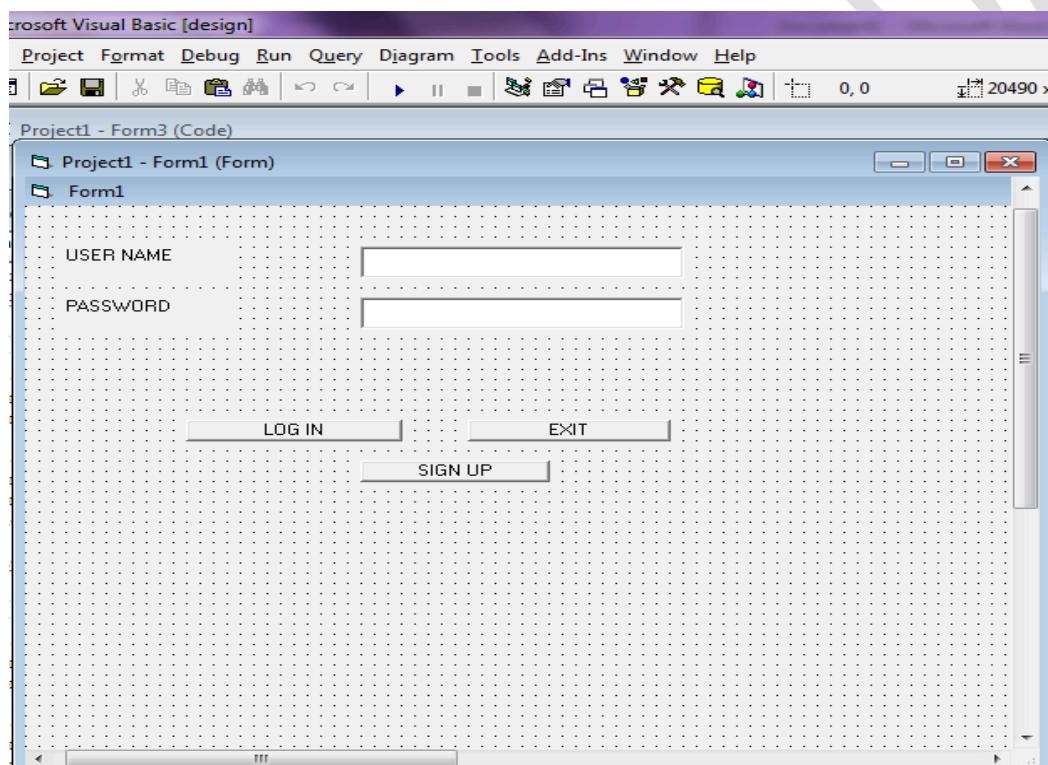


COLLABORATION DIAGRAM: for view salary slip

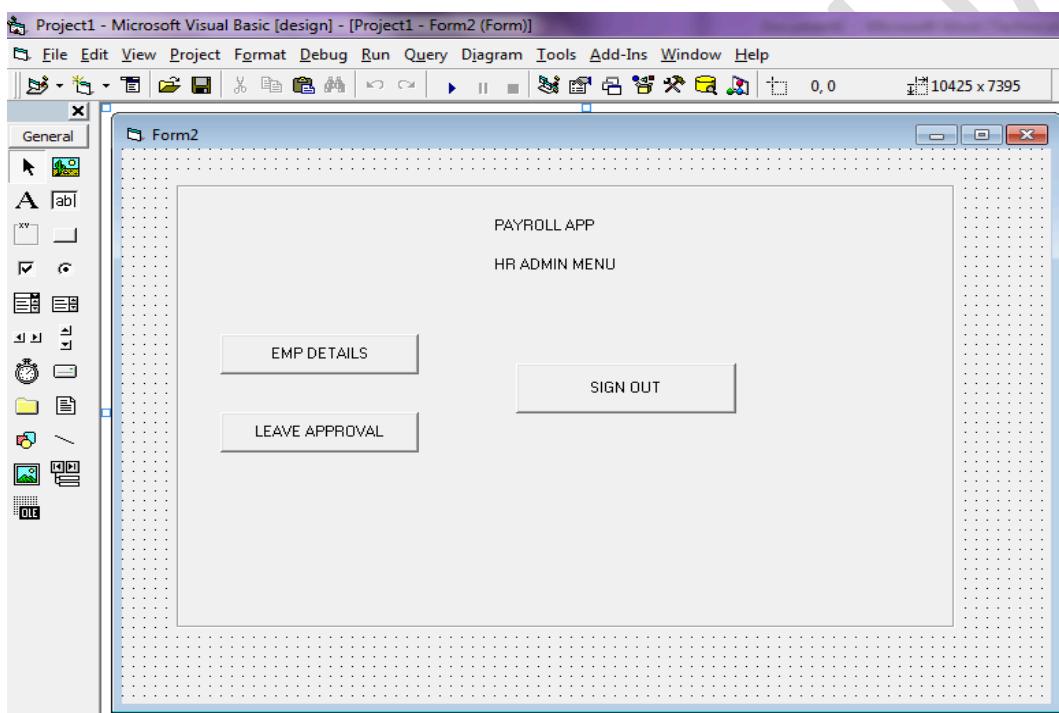


FORM DESIGN

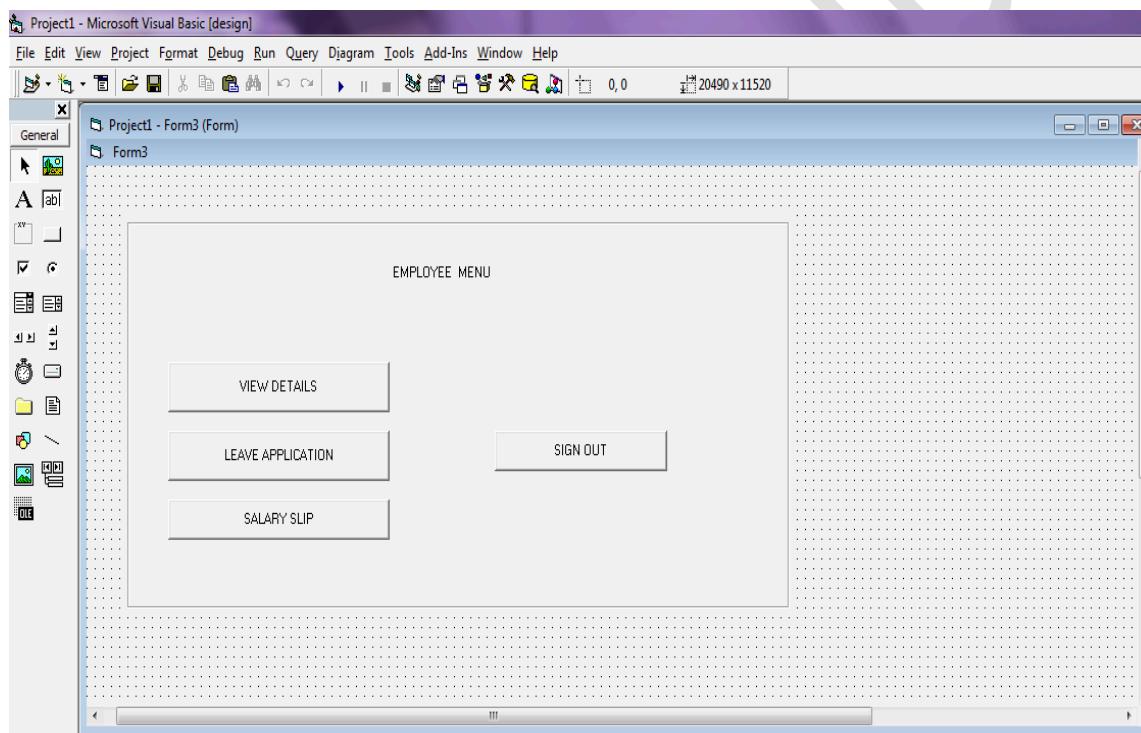
Login



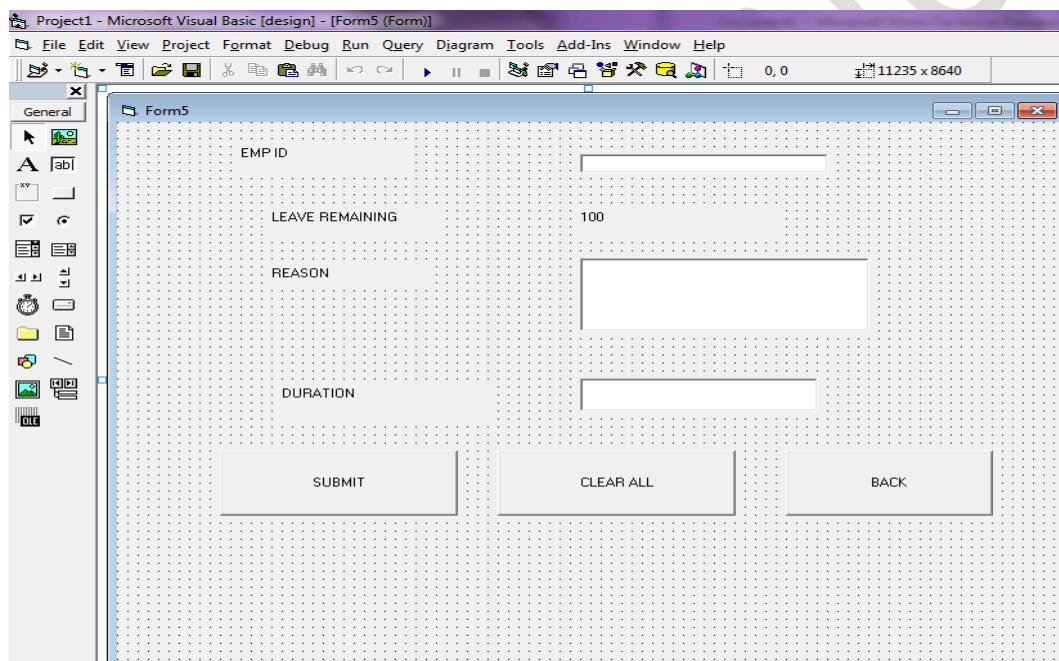
Admin Menu



Employee Menu



Leave Application



CODING

FORM1:LOGIN

```
Dim cn As ADODB.Connection  
  
Dim rs1 As ADODB.Recordset  
  
Private Sub Command1_Click()  
  
If Text1.Text = rs1(0) And Text2.Text = rs1(1) Then  
  
    MsgBox "user verified"  
  
    Form1.Visible = False  
  
    Form3.Visible = True  
  
    Form4.Command3.Visible = False  
  
    Form4.Command5.Visible = False  
  
Elseif Text1.Text = "rajeev" And Text2.Text = "rocky" Then  
  
    MsgBox "administrator verified"  
  
    Form1.Visible = False  
  
    Form2.Visible = True
```

```
Else
    MsgBox "user not verified"
    Form1.Refresh
End If
End Sub

Private Sub Command2_Click()
End
End Sub

Private Sub Command3_Click()
Form1.Visible = False
Form7.Visible = True
End Sub

Private Sub Form_Load()
Dim s As String
Set cn = New ADODB.Connection
cn.Open "dsn=user1;uid=dbsnmp;pwd=gamer"
s = "select * from user1"
Set rs1 = New ADODB.Recordset
rs1.Open s, cn, adOpenDynamic, adLockOptimistic
End Sub
```

FORM2:ADMIN MENU

```
Private Sub Command1_Click()
```

```
Form2.Visible = False  
  
Form4.Visible = True  
  
End Sub  
  
Private Sub Command2_Click()  
  
Form2.Visible = False  
  
Form6.Visible = True  
  
End Sub  
  
Private Sub Command3_Click()  
  
Form2.Visible = False  
  
Form1.Visible = True  
  
End Sub  
  
FORM3:EMPLOYEE MENU  
  
Private Sub Command1_Click()  
  
Form3.Visible = False  
  
Form4.Visible = True  
  
End Sub  
  
Private Sub Command2_Click()  
  
Form3.Visible = False  
  
Form5.Visible = True  
  
End Sub  
  
Private Sub Command3_Click()  
  
Form9.Visible = True  
  
Form3.Visible = False
```

```
End Sub

Private Sub Command4_Click()
Form3.Visible = False
Form1.Visible = True
End Sub
```

FORM4:EMPLOYEE DETAILS

```
Dim cn As ADODB.Connection
Dim rs1 As ADODB.Recordset
```

```
Private Sub Command3_Click()
rs1.AddNew
rs1(0) = Text1.Text
rs1(1) = Text2.Text
rs1(2) = Text3.Text
rs1(3) = Text4.Text
rs1(4) = Text5.Text
rs1(5) = Text6.Text
rs1(6) = Text7.Text
rs1(7) = Text8.Text
rs1(8) = Text9.Text
rs1.Update
MsgBox "all the details is saved"
```

```
End Sub

Private Sub Command4_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
Text9.Text = ""

End Sub

Private Sub Command5_Click()
Form4.Visible = False
Form8.Visible = True
End Sub

Private Sub Command6_Click()
If Form1.Text1.Text = "rajeev" Then
Form4.Visible = False
Form2.Visible = True
Else
Form4.Visible = False
Form3.Visible = True
End If
End Sub
```

End If

End Sub

Private Sub Command7_Click()

Dim a, b As String

a = InputBox("enter emp id")

b = Val(a)

Set rs1 = cn.Execute("select * from empd1 where eid=" & b)

If rs1.EOF <> True Then

Text1.Text = rs1(0)

Text2.Text = rs1(1)

Text3.Text = rs1(2)

Text4.Text = rs1(3)

Text5.Text = rs1(4)

Text6.Text = rs1(5)

Text7.Text = rs1(6)

Text8.Text = rs1(7)

Text9.Text = rs1(8)

Else

MsgBox "record doesn't exist"

End If

End Sub

Private Sub Command8_Click()

Text9.Text = Val(Text5.Text) + Val(Text6.Text) + Val(Text7.Text) - Val(Text8.Text)

```
End Sub

Private Sub Form_Load()
    Dim s As String
    Set cn = New ADODB.Connection
    cn.Open "dsn=empd1;uid=dbsnmp;pwd=gamer"
    s = "select * from empd1"
    Set rs1 = New ADODB.Recordset
    rs1.Open s, cn, adOpenDynamic, adLockOptimistic
End Sub
```

FORM5:LEAVE APPLICATION

```
Dim cn As ADODB.Connection
Dim rs As ADODB.Recordset
Private Sub Command1_Click()
    Dim d As String
    d = Val(Text3.Text)
    rs.AddNew
    rs.MoveFirst
    rs.Find "eid=" & d
    rs(9) = Label3.Caption
    rs(10) = Text1.Text
    rs(11) = Text2.Text
```

```
rs.Update  
MsgBox "detail submitted"  
End Sub  
  
Private Sub Command2_Click()  
Text1.Text = ""  
Text2.Text = ""  
Text3.Text = ""  
End Sub
```

```
Private Sub Command3_Click()  
Form5.Visible = False  
Form4.Visible = True  
End Sub  
  
Private Sub Form_Load()  
Dim s3 As String  
Set cn = New ADODB.Connection  
cn.Open "dsn=empd1;uid=dbsnmp;pwd=gamer"  
s3 = "select * from empd1"  
Set rs = New ADODB.Recordset  
rs.Open s3, cn, adOpenDynamic, adLockOptimistic  
End Sub
```

FORM6:LEAVE APPROVAL/REJECT

```
Dim cn As ADODB.Connection
```

```
Dim rs1 As ADODB.Recordset
```

```
Private Sub Command1_Click()
```

```
If Val(Text3.Text) < 100 Then
```

```
    MsgBox "approved"
```

```
    Command2.Enabled = True
```

```
    Command1.Enabled = False
```

```
Else
```

```
    MsgBox "can't be approved"
```

```
    Command2.Enabled = True
```

```
    Command1.Enabled = False
```

```
End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
If Val(Text3.Text) > 100 Then
```

```
    MsgBox "rejected"
```

```
    Command2.Enabled = False
```

```
    Command1.Enabled = True
```

```
Else
```

```
    MsgBox "can't rejected"
```

```
    Command2.Enabled = False
```

```
    Command1.Enabled = True
```

```
End If
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Dim e As String
```

```
e = Val(Text1.Text)
```

```
rs1.MoveFirst
```

```
rs1.Find "eid=" & e
```

```
Text1.Text = rs1(0)
```

```
Text2.Text = rs1(10)
```

```
Text3.Text = rs1(11)
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Form6.Visible = False
```

```
Form2.Visible = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim s2 As String
```

```
Set cn = New ADODB.Connection
```

```
cn.Open "dsn=empd1;uid=dbsnmp;pwd=gamer"
```

```
s2 = "select * from empd1"
```

```
Set rs1 = New ADODB.Recordset
```

```
rs1.Open s2, cn, adOpenDynamic, adLockOptimistic
```

```
End Sub
```

FORM7:SIGNUP

```
Dim cn As ADODB.Connection
```

```
Dim rs1 As ADODB.Recordset
```

```
Private Sub Command1_Click()
```

```
    rs1.AddNew
```

```
    rs1(0) = Text1.Text
```

```
    rs1(1) = Text2.Text
```

```
    rs1.Update
```

```
    MsgBox "data is saved"
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    Form7.Visible = False
```

```
    Form1.Visible = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Dim s As String
```

```
    Set cn = New ADODB.Connection
```

```
    cn.Open "dsn=user1;uid=dbsnmp;pwd=gamer"
```

```
    s = "select * from user1"
```

```
    Set rs1 = New ADODB.Recordset
```

```
    rs1.Open s, cn, adOpenDynamic, adLockOptimistic
```

```
End Sub
```

FORM8:DELETE EMP DETAILS

```
Dim cn As ADODB.Connection
```

```
Dim rs1 As ADODB.Recordset
```

```
Private Sub Command1_Click()
```

```
Dim aa As String
```

```
aa = Text1.Text
```

```
rs1.MoveFirst
```

```
rs1.Find ("eid=" & aa)
```

```
rs1.Delete
```

```
rs1.Update
```

```
MsgBox "record deleted"
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form8.Visible = False
```

```
Form4.Visible = True
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim s1 As String
```

```
Set cn = New ADODB.Connection
```

```
cn.Open "dsn=empd1;uid=dbsnmp;pwd=gamer"
```

```
s1 = "select * from empd1"
```

```
Set rs1 = New ADODB.Recordset
```

```
rs1.Open s1, cn, adOpenDynamic, adLockOptimistic
```

End Sub

FORM9:SALARY SLIP

Dim cn As ADODB.Connection

Dim rs1 As ADODB.Recordset

Private Sub Command1_Click()

Dim ab As String

ab = InputBox("enter employee id")

Set rs1 = cn.Execute("select * from empd1 where eid= " & ab)

If rs1.EOF <> True Then

Label4.Caption = rs1(0)

Label5.Caption = rs1(1)

Label6.Caption = rs1(8)

Else

MsgBox "record doesn't exist"

End If

End Sub

Private Sub Command2_Click()

Form9.Visible = False

Form3.Visible = True

End Sub

Private Sub Form_Load()

Dim s2 As String

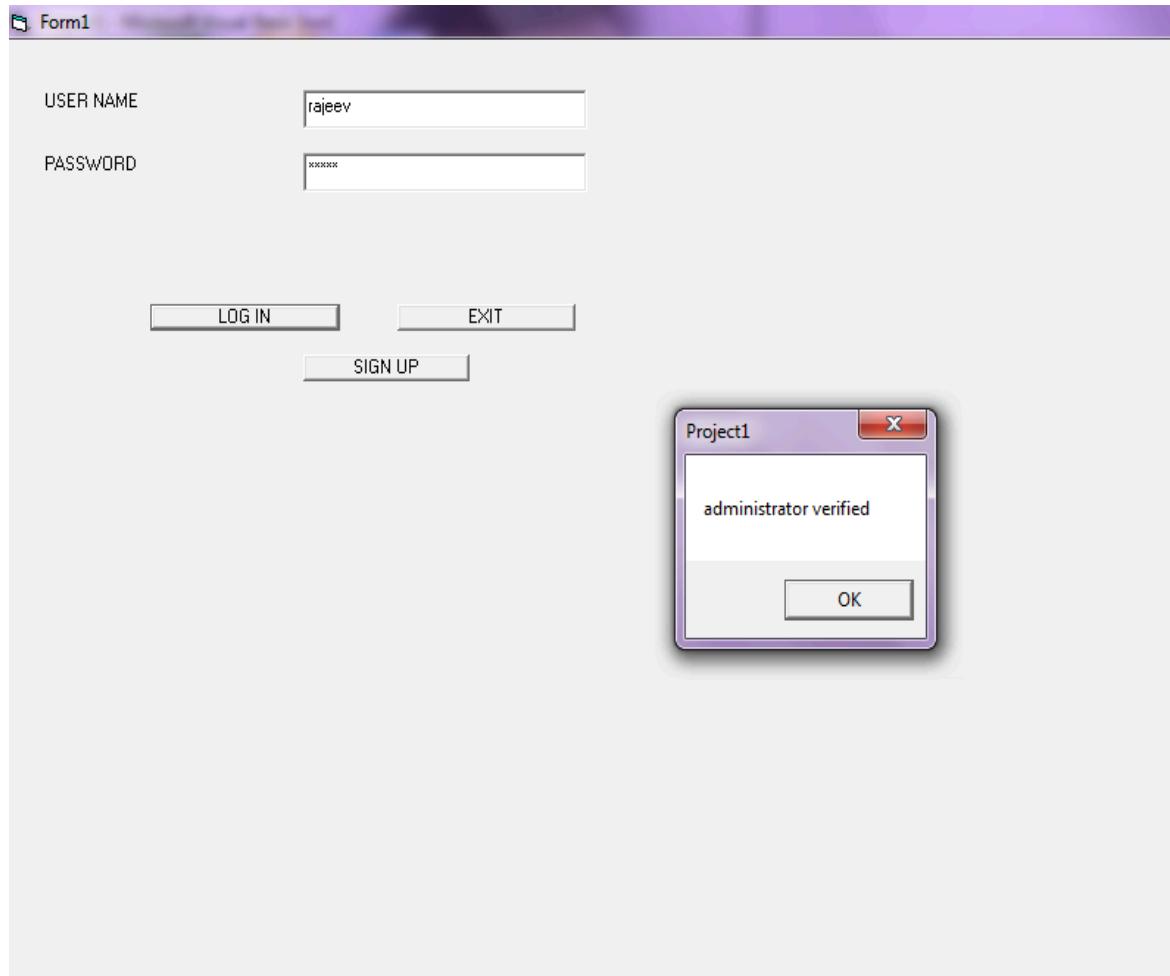
Set cn = New ADODB.Connection

```
cn.Open "dsn=empd1;uid=dbsnmp;pwd=gamer"  
s2 = "select * from empd1"  
Set rs1 = New ADODB.Recordset  
rs1.Open s2, cn, adOpenDynamic, adLockOptimistic  
End Sub
```

STAFF MANUAL

OUTPUT

LOGIN FORM:

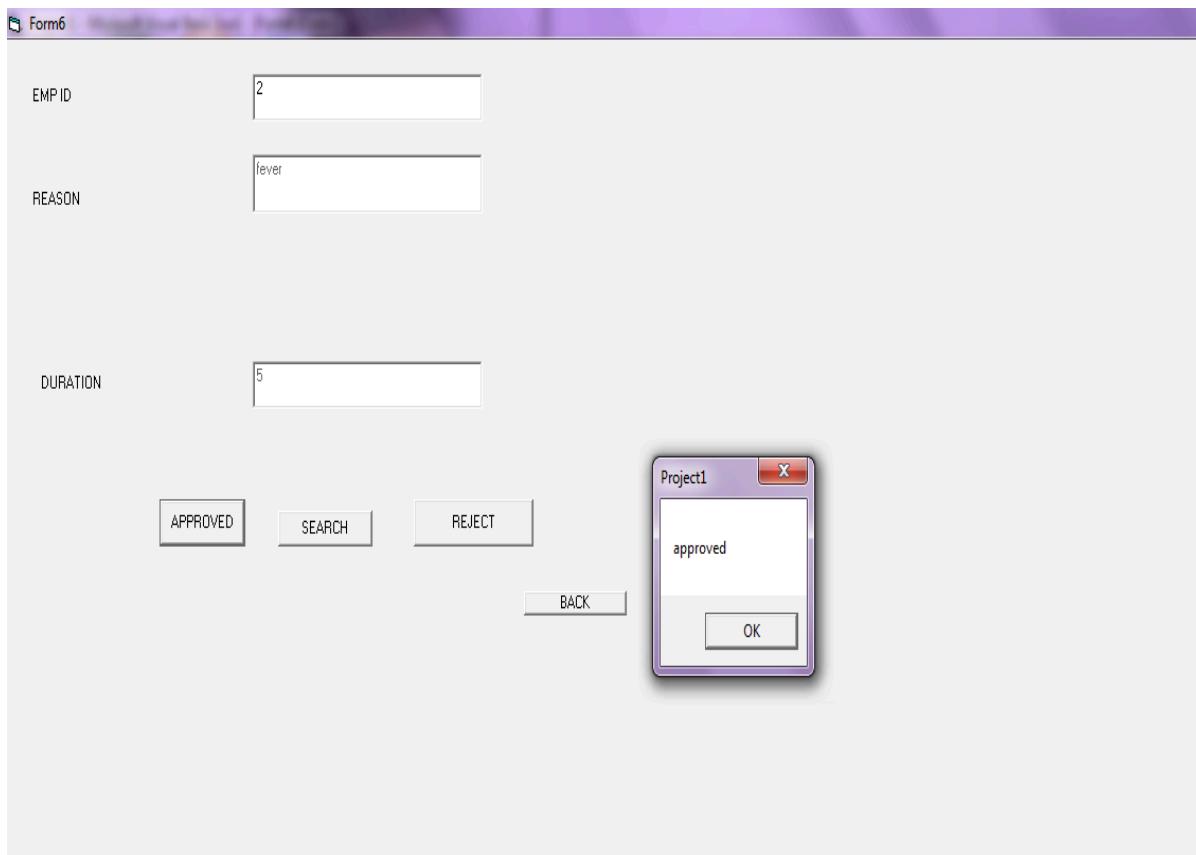


ADD EMPLOYEE DETAILS:

EMPLOYEE DETAILS

EMP_ID	<input type="text" value="8"/>	DEPARTMENT	<input type="text" value="fgf"/>
NAME :	<input type="text" value="ravi"/>	DESIGNATION	<input type="text" value="hr"/>
CATEGORY		AMOUNT	
BASIC SALARY		<input type="text" value="45445"/>	Project1
D.A		<input type="text" value="2272.3"/>	 all the details is saved
H.R.A		<input type="text" value="4544.5"/>	 OK
GROSS AMT		<input type="text" value="52261.8"/>	
TAX		<input type="text" value="1045.235"/>	
NET SALARY		<input type="text" value="51216.515"/>	

LEAVE APPROVAL FORM:



Result:

Thus the payroll system is designed and implemented successfully and the output is verified.

Ex.No.4

INVENTORY SYSTEM

AIM

To develop the Inventory system System using rational rose tools, visual basic and MS access.

Problem analysis and project planning

Introduction

The stock maintenance system is basically for the customers who access the information about the stock (here it is books in the book store) and retrieves the information.

Objectives

The purpose of the document is to define the requirements of the stock maintenance system. This supplementary specification lists the requirements that are not readily captured in the use cases of the use case model. The supplementary specification & the use case model together capture a complete set of requirement on the system.

Scope

This supplementary specification applies to the stock maintenance system. This specification defines the non-functional requirements of the system, such as reliability, usability, performance and supportability as well as functional requirements that are common across a number of use cases.

Problem Statement

A new stock maintenance system for a book store is to replace the existing maintenance system which is inefficient. The new stock maintenance system will allow the employee to record information of the books available in the book store and generate report based on the total amount of sales.

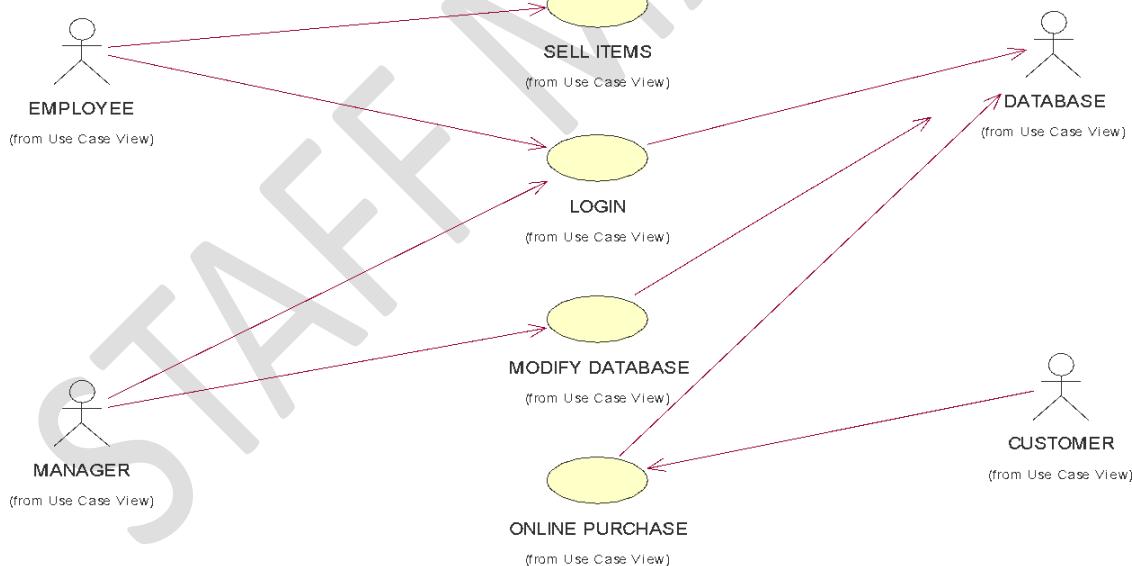
The new system will have a Windows based desktop interface to allow employees to enter the information of sale, purchase orders, change employee preferences and create reports. Employees can only access the information and purchase orders for security purpose.

The system retains information on all the books in the shop. The system retains the records of the cost, edition, author, publication of the books. The employee maintains the information of the sale of books. He can add the books at right time and update the database.

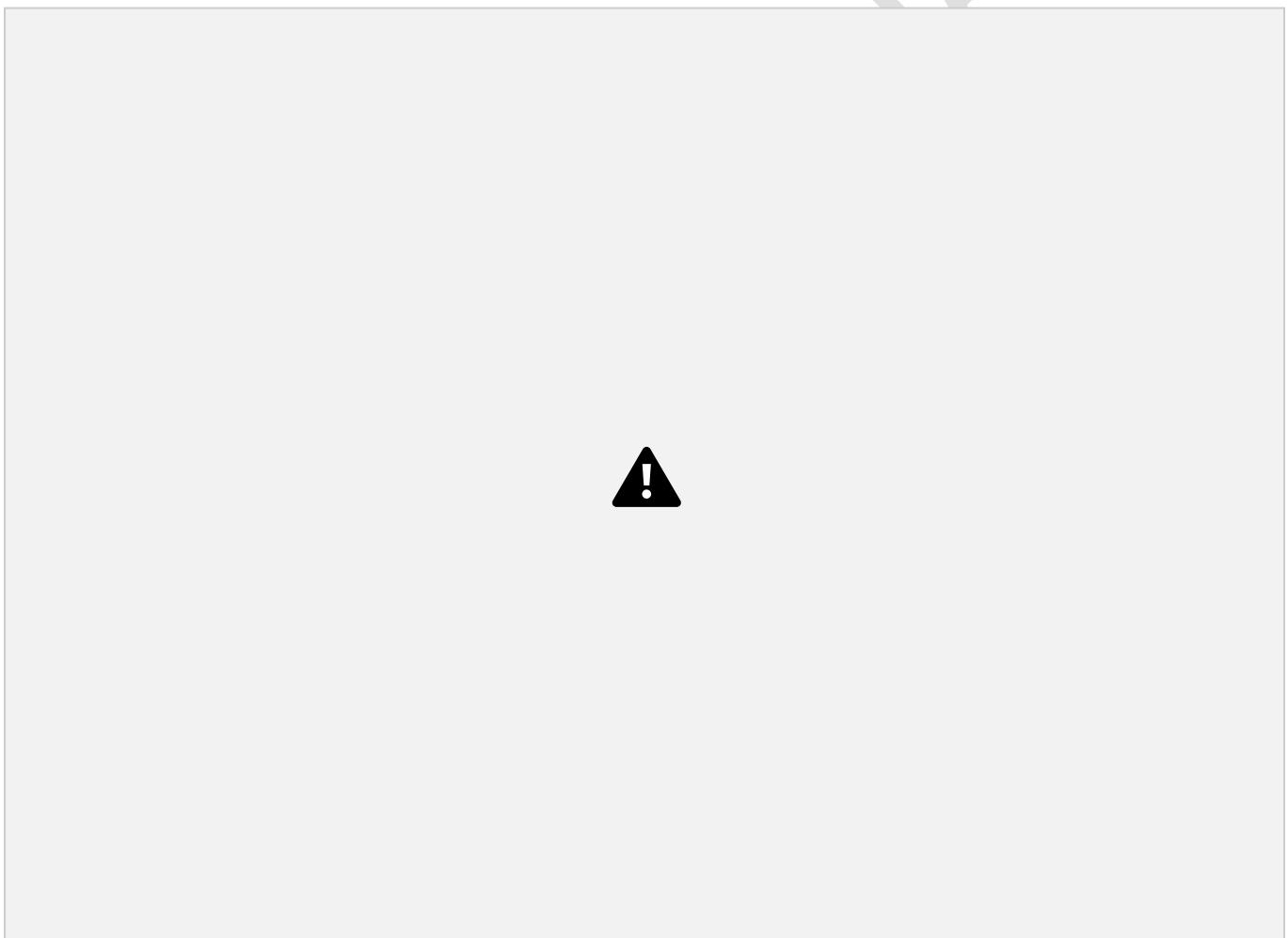
The customer can view the availability of the required books and the price of the books. The customer can just view them but cannot make any changes.

UML Diagrams:

Use Case Diagram



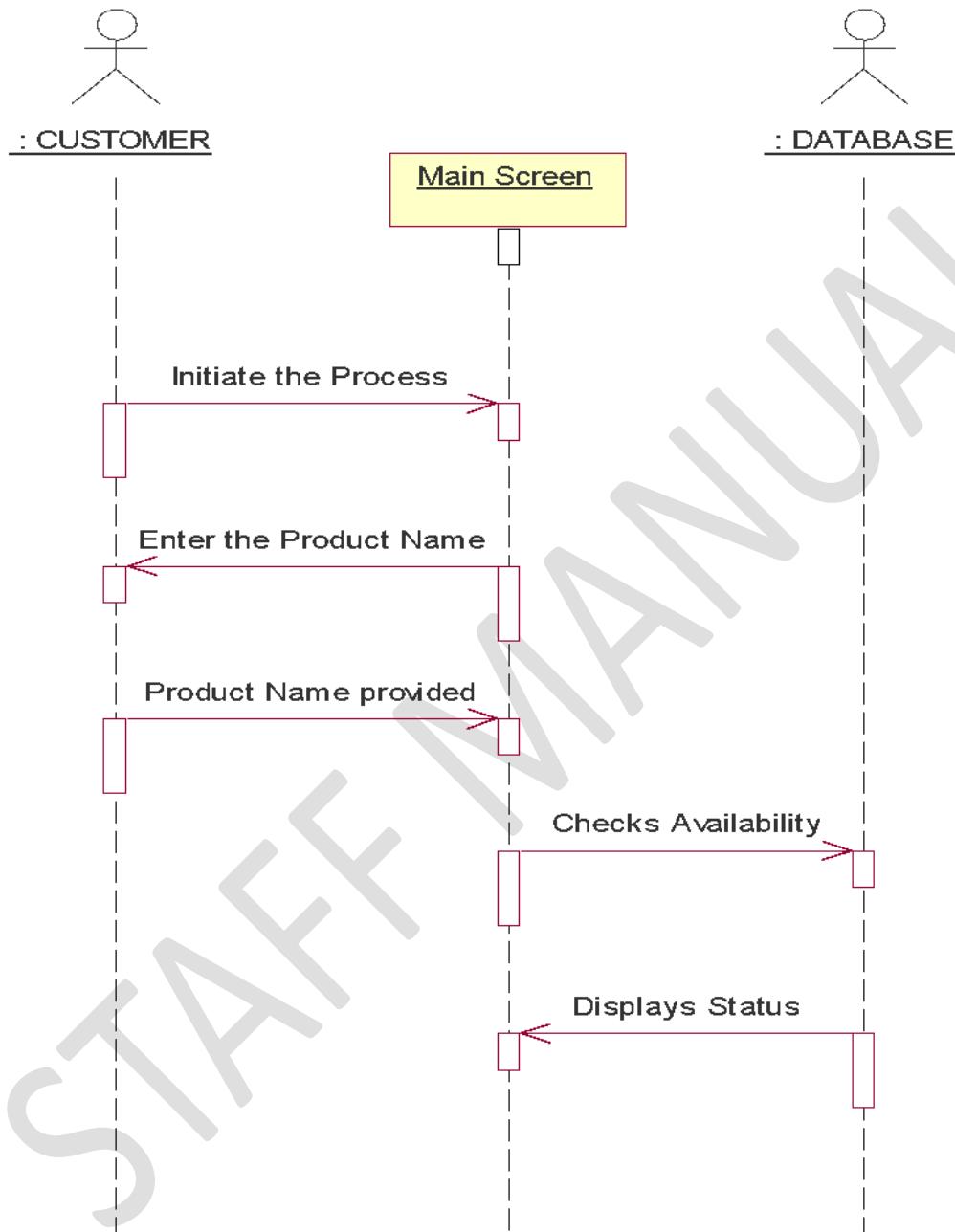
Activity Diagram:



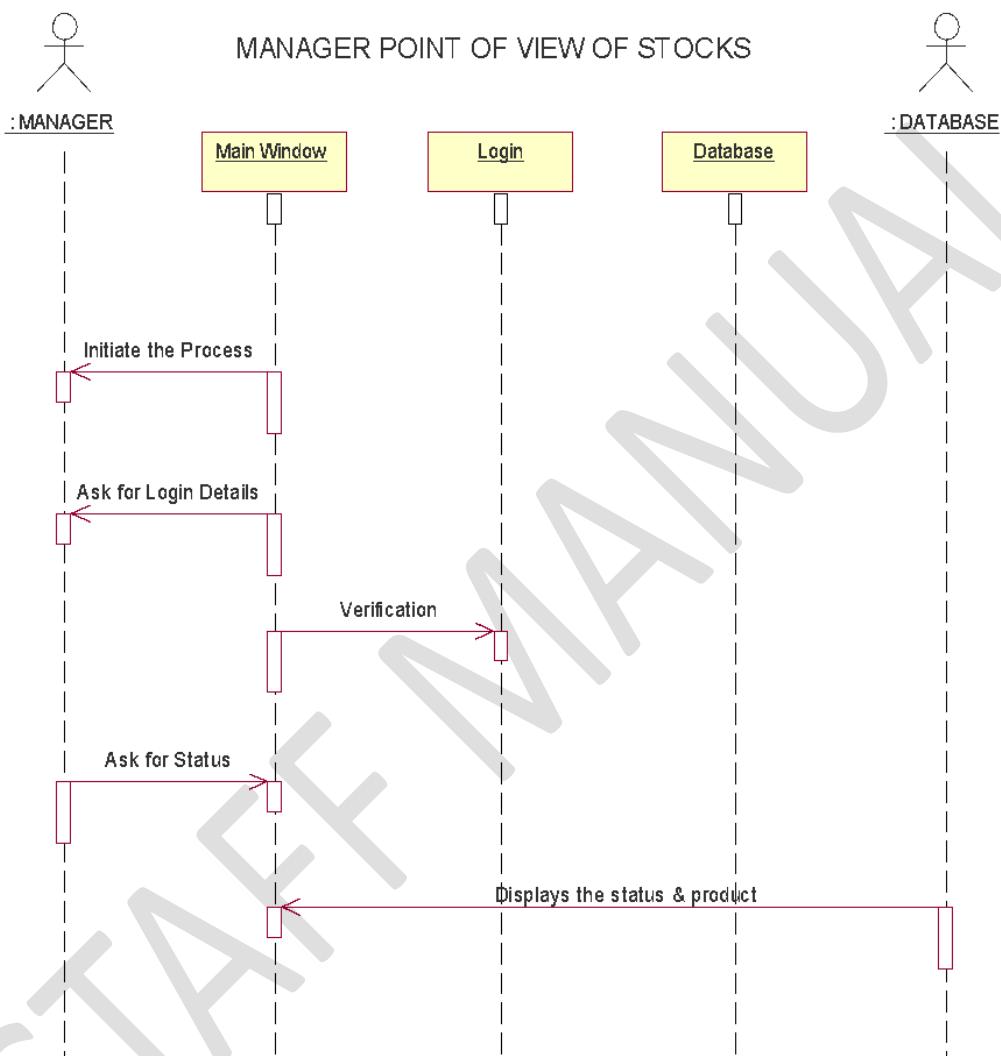
Sequence Diagrams:

Availability of Stocks

STAFF MANUAL

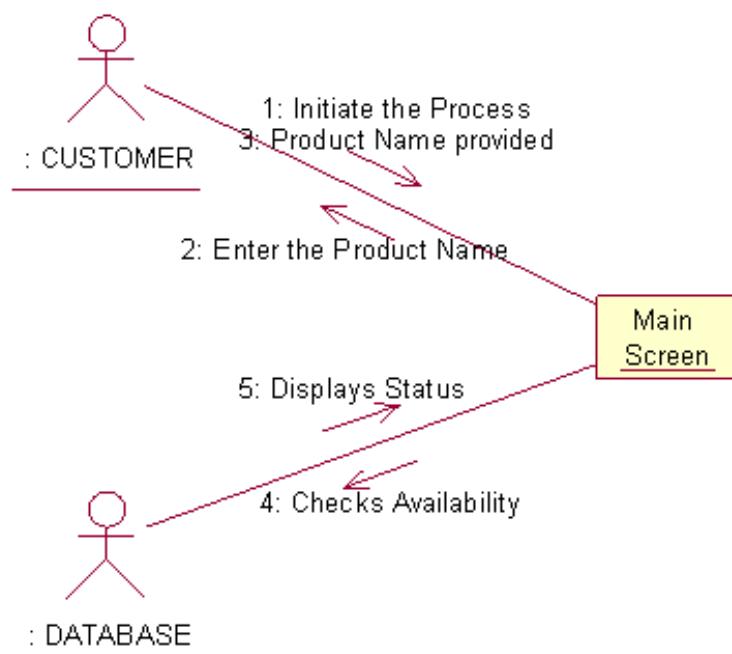


Manager Point of View

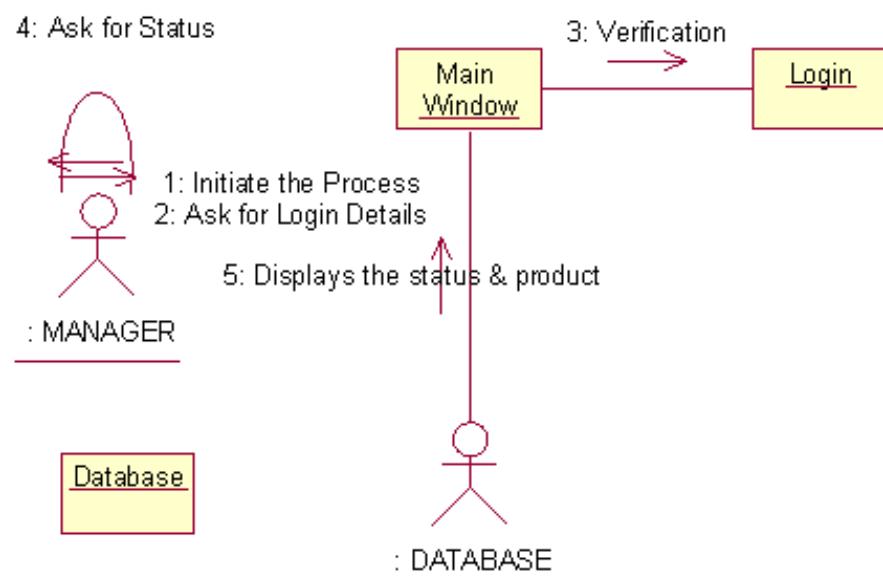


Collaboration Diagrams:

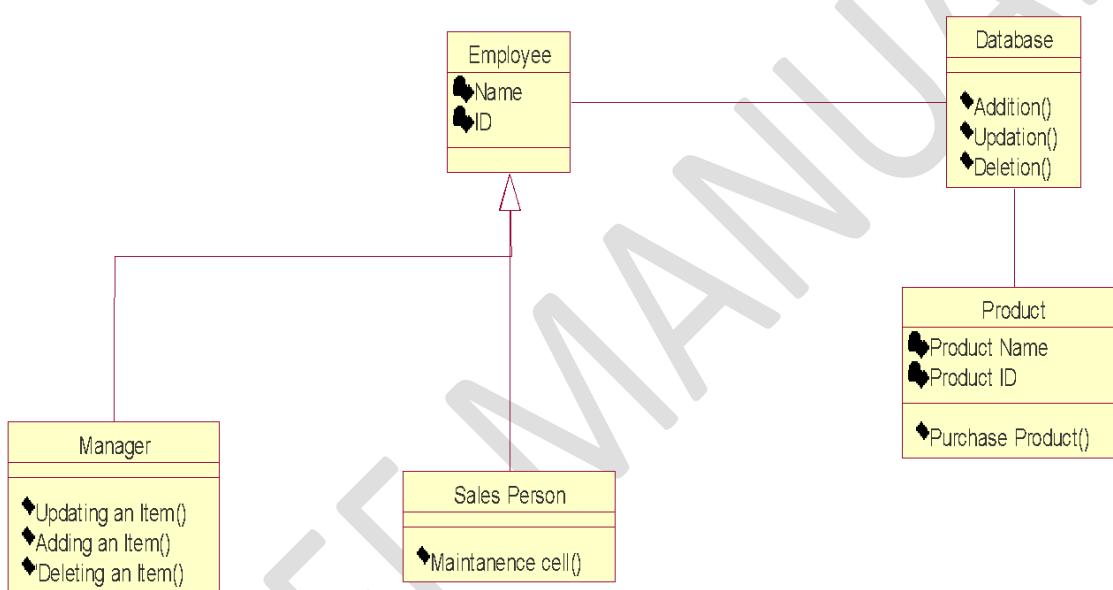
Availability of Stocks



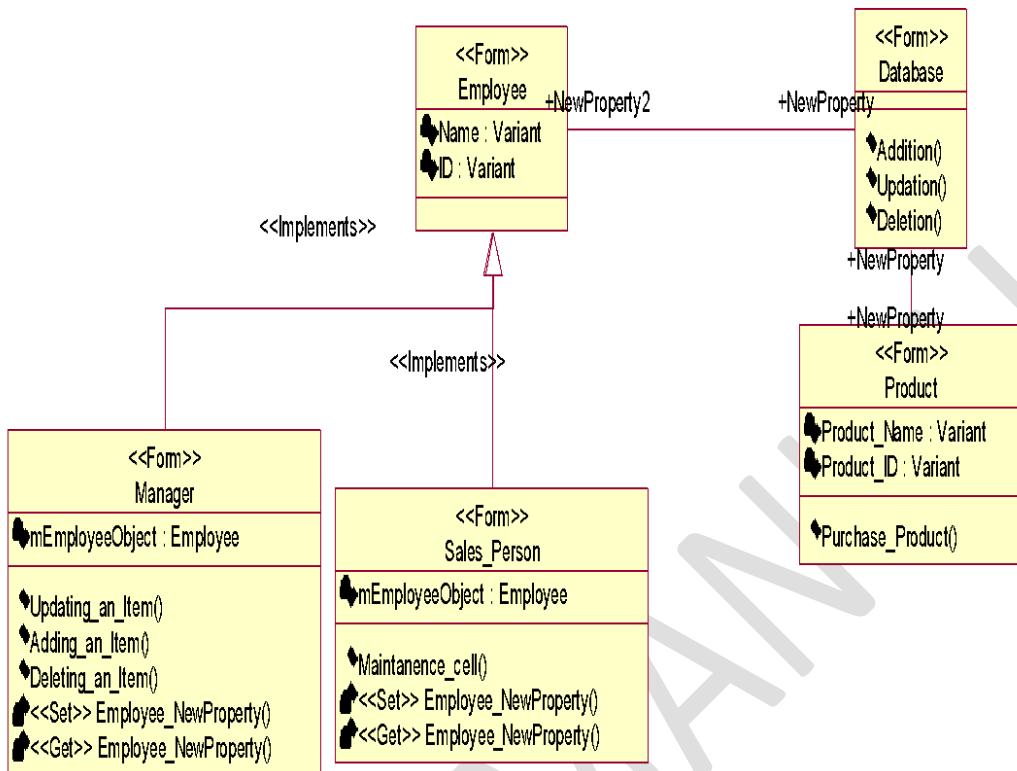
Manager Point of View



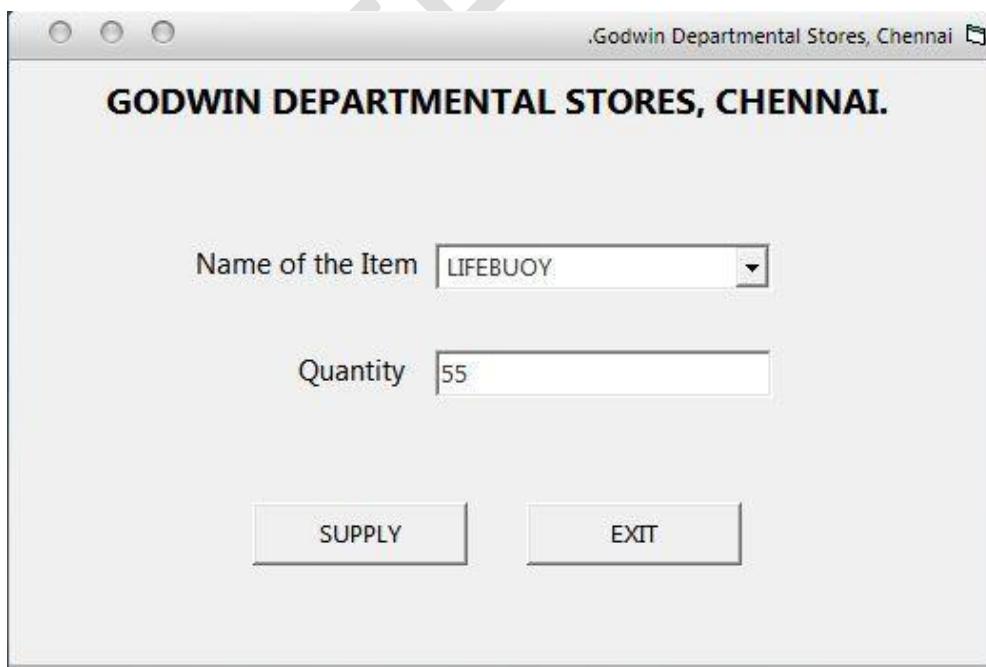
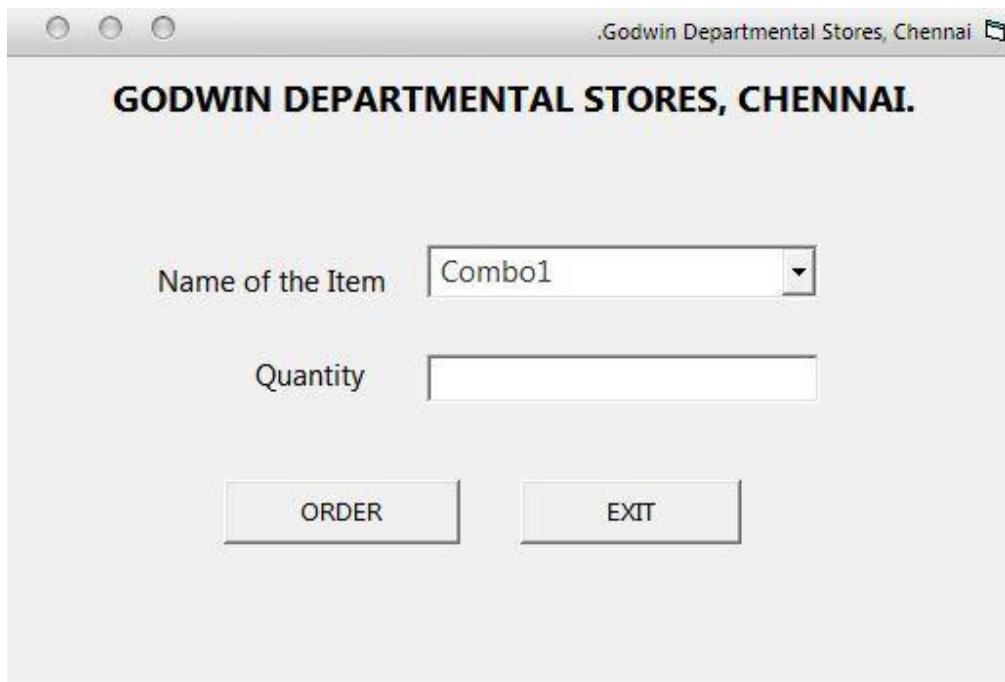
Class Diagram:



Forward Engineering Class Diagram:



Form Design:



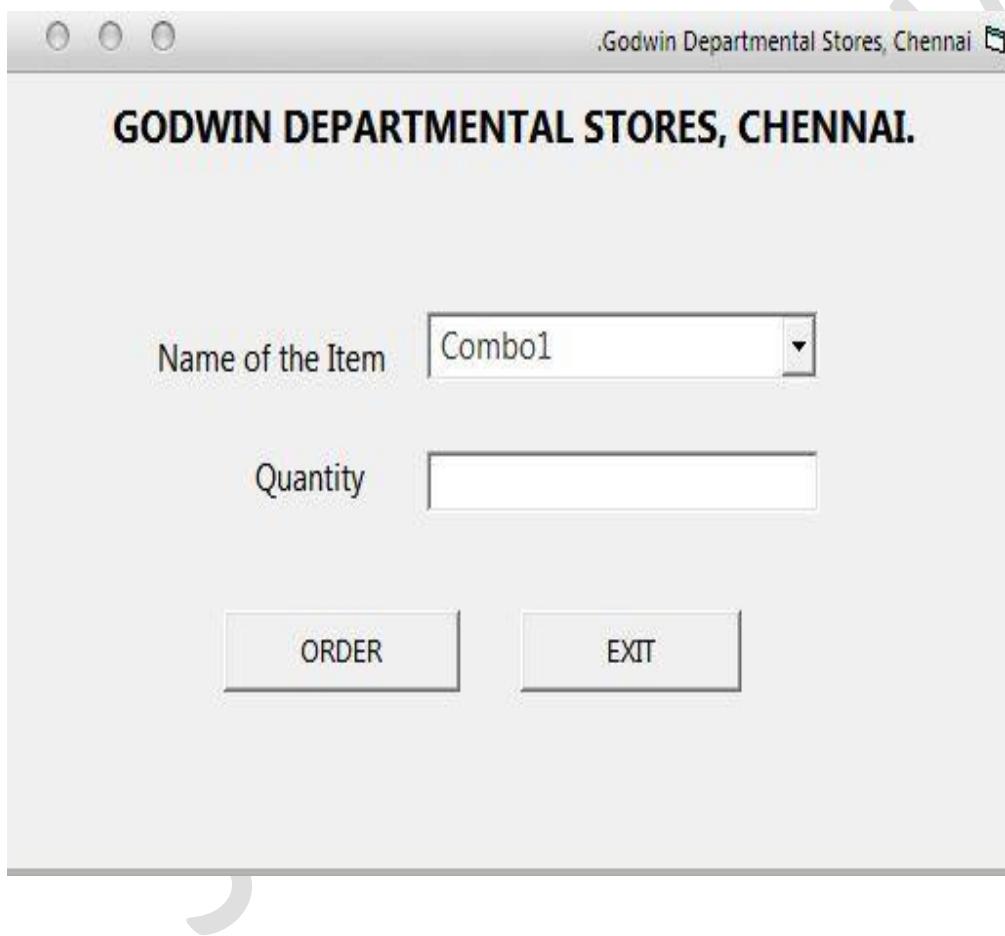
Coding:

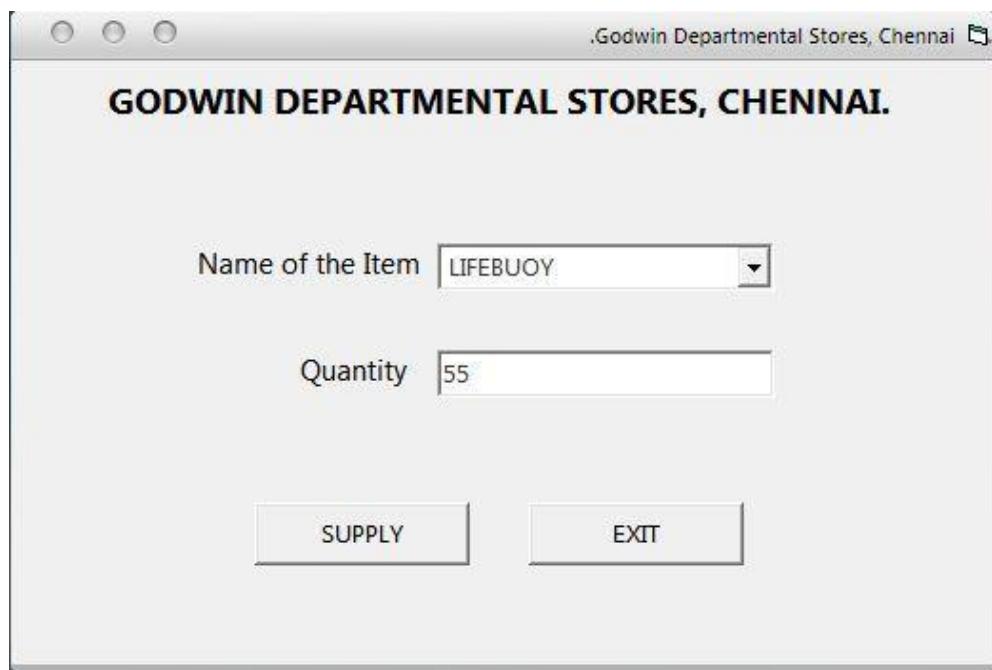
```
Private Order_click()  
  
If Combo1.Text = Combo1 And Text1.Text = "" Then  
  
MsgBox"Enter name and quantity"  
  
Combo1.SetFocus  
  
Else: If Val(Text1.Text) > 500 Then MsgBox"scarcity of items"  
  
End  
  
Else  
  
Form2.Show  
  
End If  
  
End Sub  
  
Private Sub Exit_click()  
  
End  
  
End Sub  
  
Private Sub Form_load()  
  
Combo1.AddItem "oil"  
  
Combo1.AddItem "shampoo"  
  
Combo1.AddItem "Printer"  
  
Combo1.AddItem "mouse"  
  
End Sub
```

Form 2

```
Private Sub Exit_Click()  
    MsgBox "Thank You, Visit Again! ☺"  
End  
End Sub  
  
Private Sub Form_Load()  
    Text1.Text = Form1.Text1.Text  
    Combo1.Text = Form1.Combo1.Text  
End Sub  
  
Private Sub Supply_Click()  
    MsgBox "Congrats, Your Order has been Sanctioned!"  
    Data1.Recordset.AddNew  
    Data1.Recordset.Fields(0) = Combo1.Text  
    Data1.Recordset.Fields(1) = Text1.Text  
    Data1.Recordset.Update  
End Sub
```

OUTPUT:





Congrats, Product Delivered!

OK

Result:

Thus the Stock maintenance Project is analyzed and designed using Rational Rose and Implemented in VB.

EX.NO :5

QUIZ SYSTEM

Problem analysis and project planning

Introduction

Quiz systems are generally used to test the knowledge of an individual person. But persons in technical arena should be well stuffed in both technical and non-technical sides. So this system is basically used to test the professionals.

Objectives

The main objective of this quiz system is to replace the existing hand written system with a fully computerized system that would easily and exactly calculate the intellectual capabilities of a person.

Scope

Earlier hand written test were conducted and it was tedious and time consuming to correct the papers. Even it had a greater probability of errors. So the scope of this system is that it will ease the trends followed in current quiz system.

Problem Statement

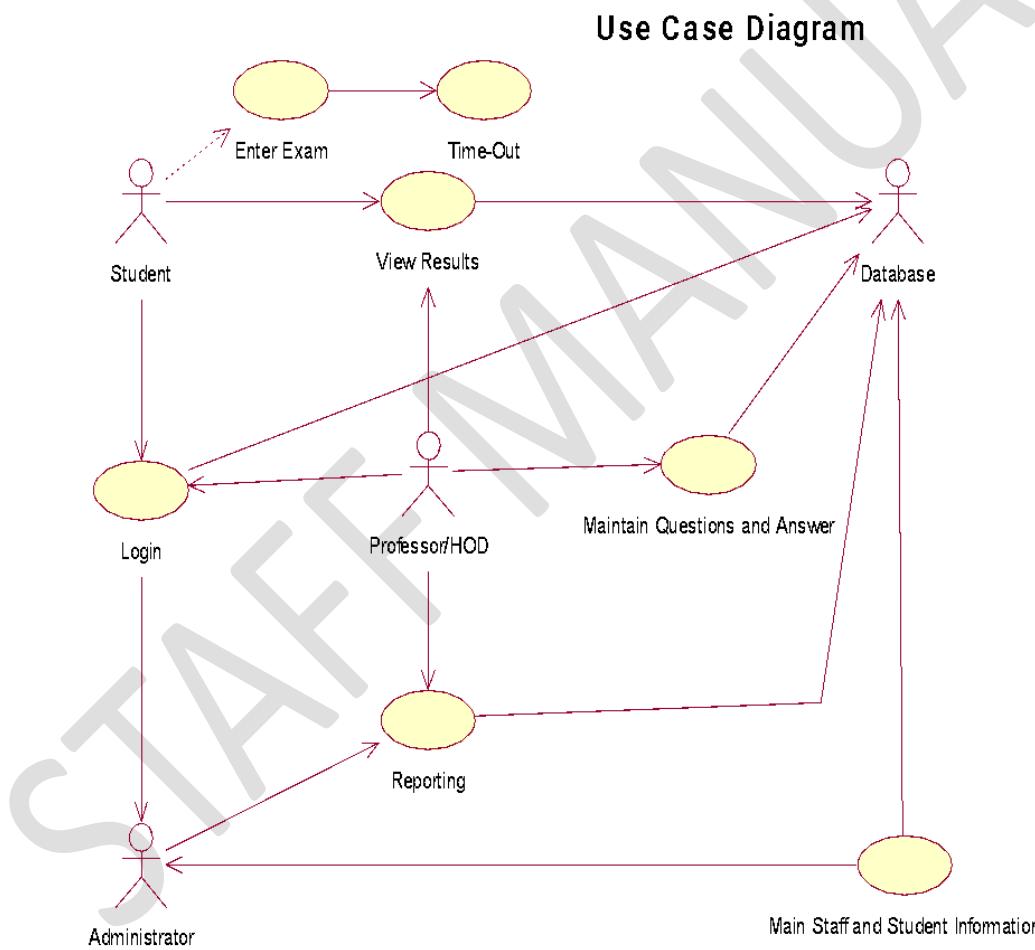
The project requires an efficient quiz system to access the students of third year IT, based on their skills in various subjects. It generates report of all the students who have taken up the quiz and stores it for future use. Access rights are allocated in following order.

Administrator->Professors/H.O.D->Students

The administrator is given rights only to add or delete student profiles. But has no rights to access the quiz information. Professors and lecturers are given rights to access the question available and if needed, make changes to them. They can also access the report database to view the overall and individual performance of the students and then take the necessary action. All the reports and queries are at their disposal.

Students are the end users of the system. They attend the quiz session and their marks are added to the database, final reporting is done based on their performances. The student cannot access any of the databases including the questions, report or the profile database. Each student is given a username and password to ensure the security.

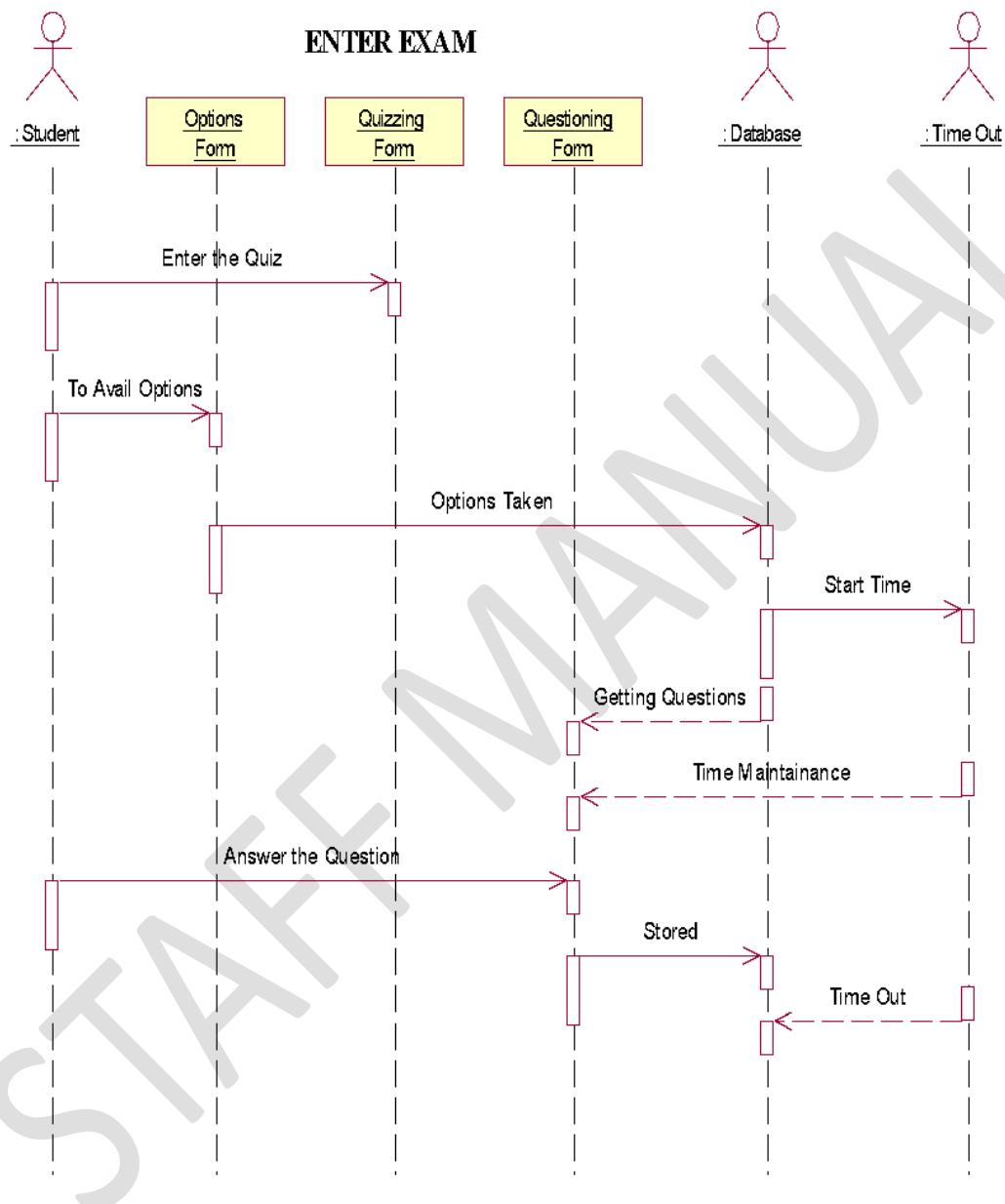
UML Diagram

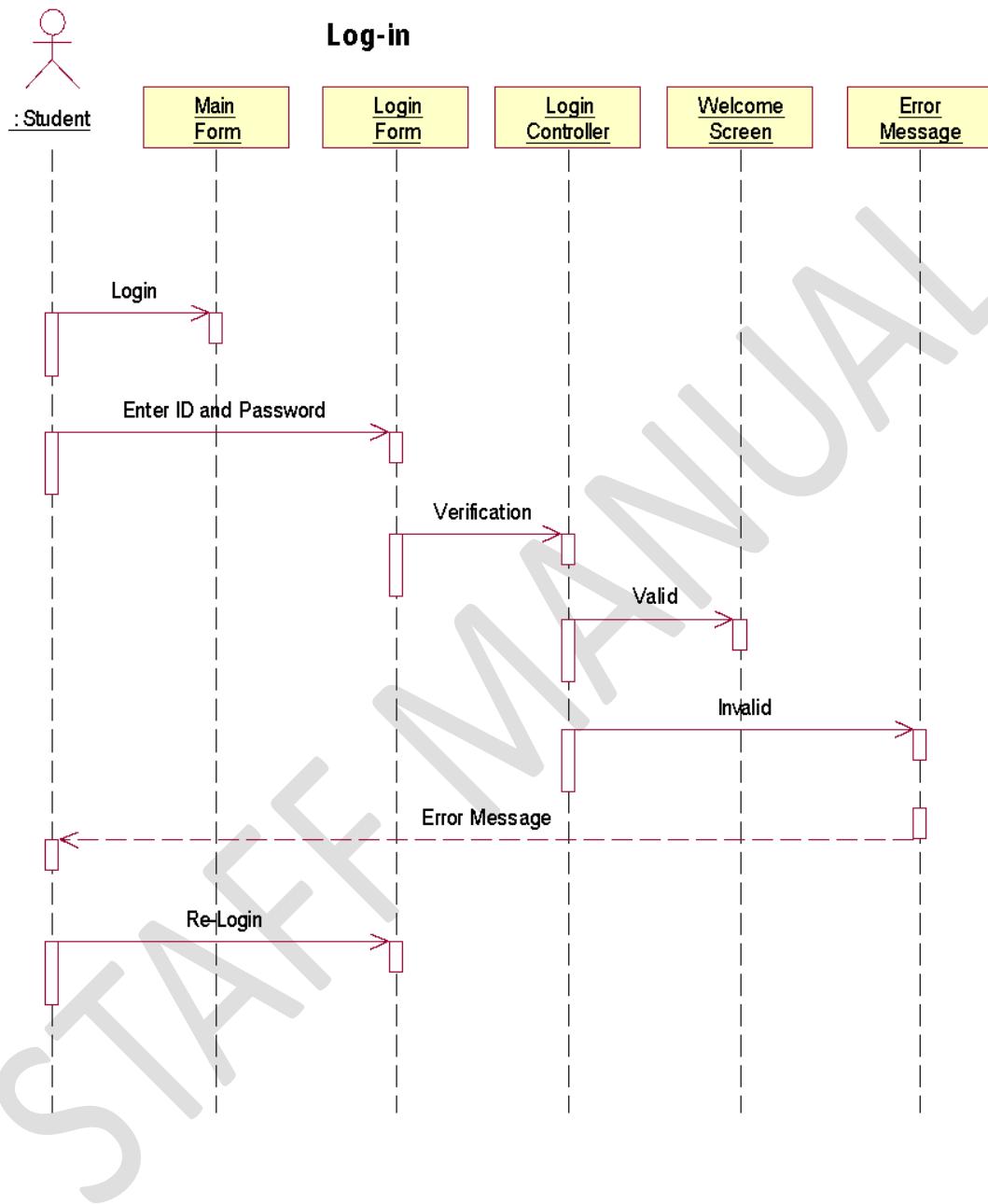


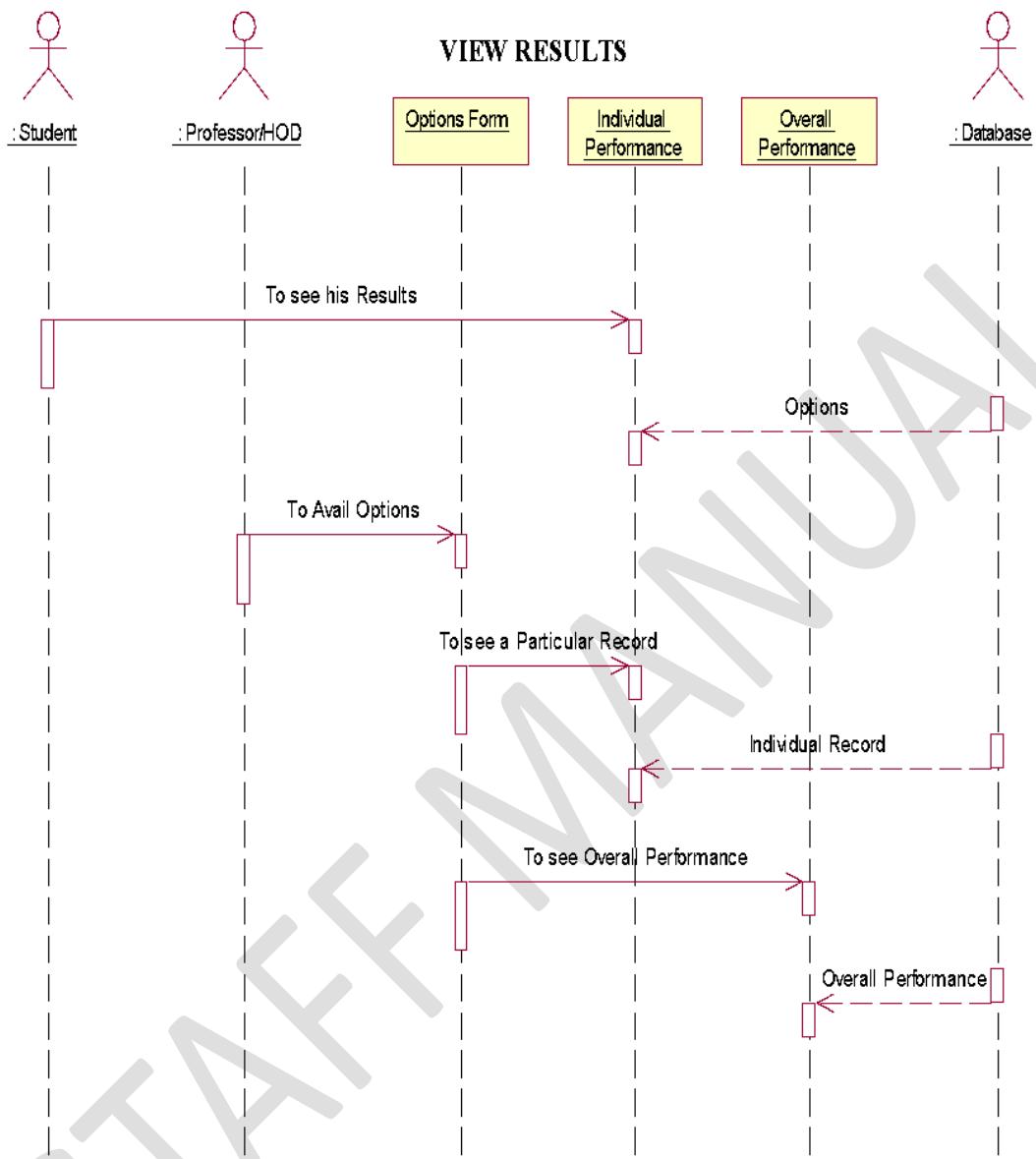
Sequence Diagrams



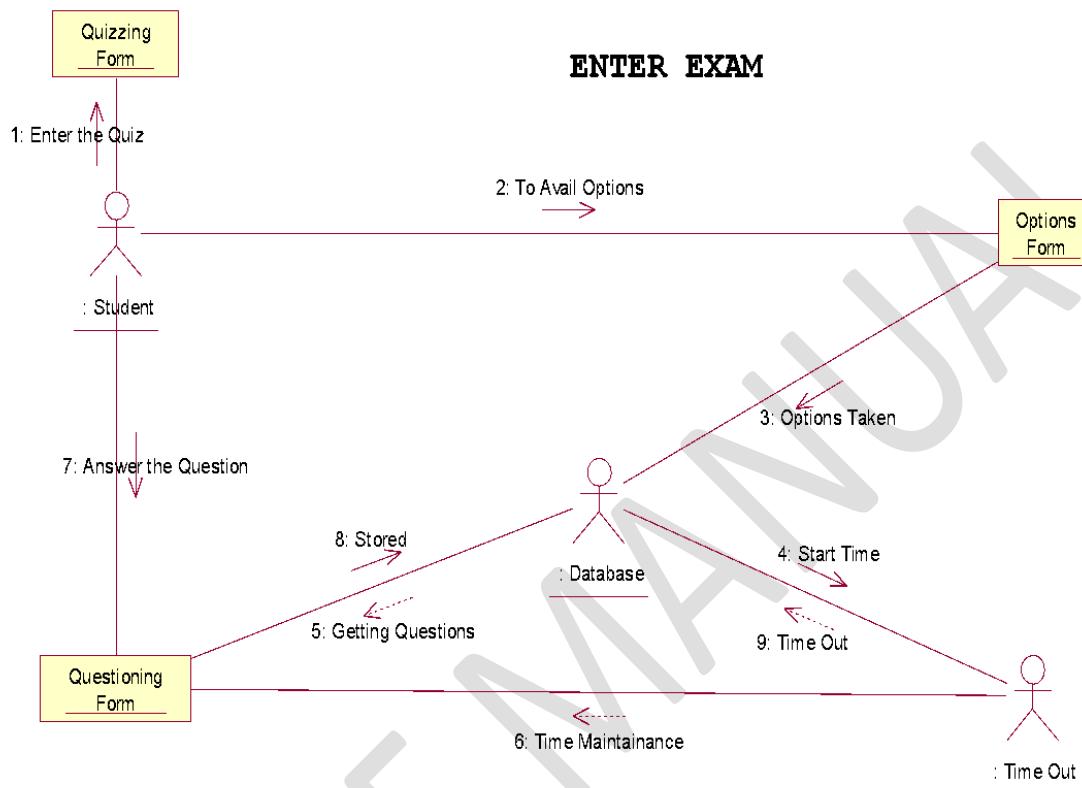
Sequence Diagram

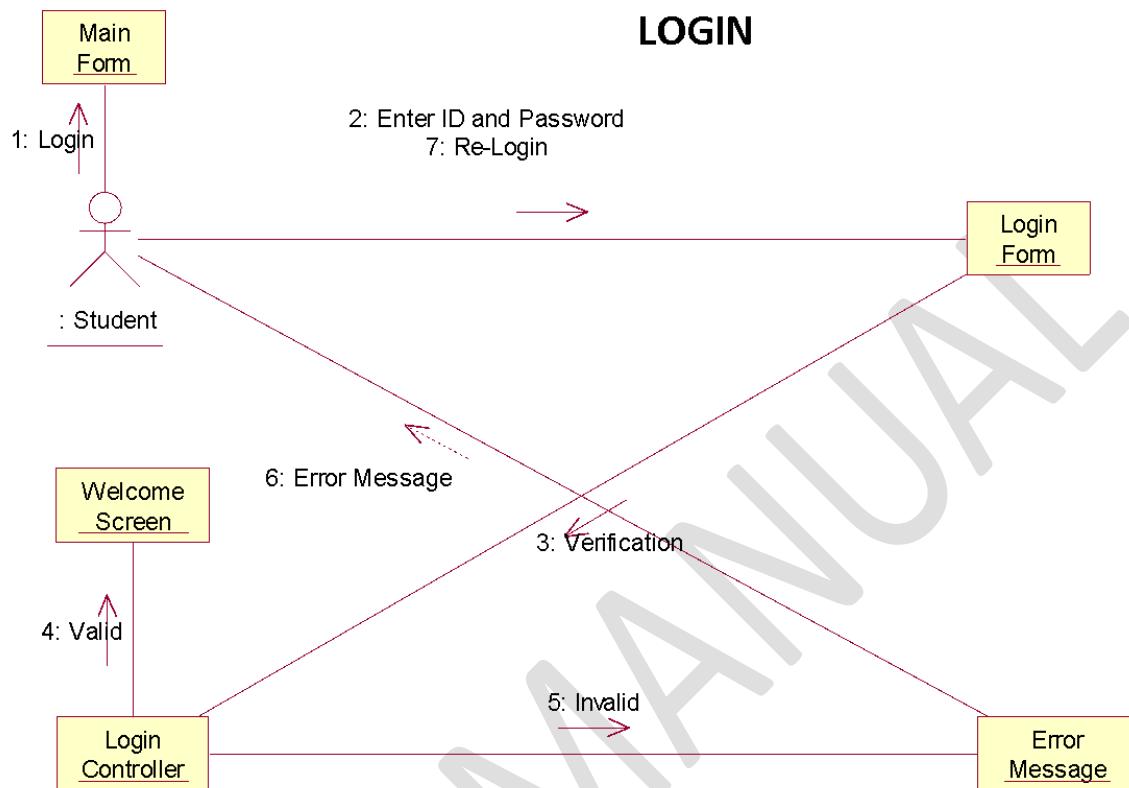


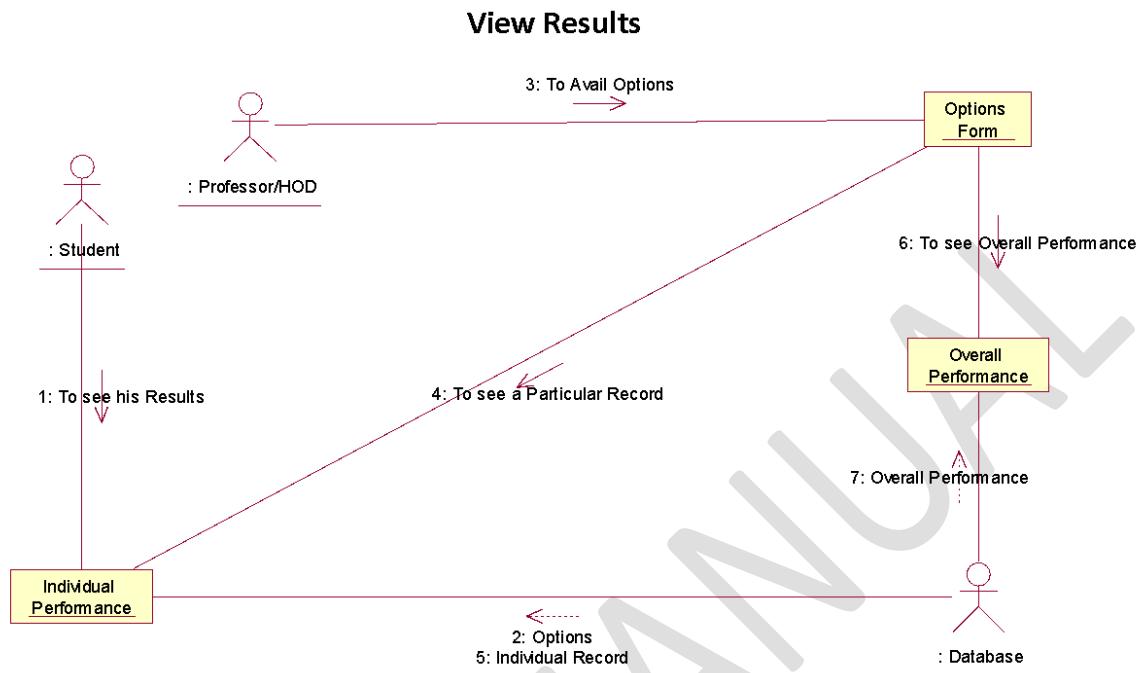




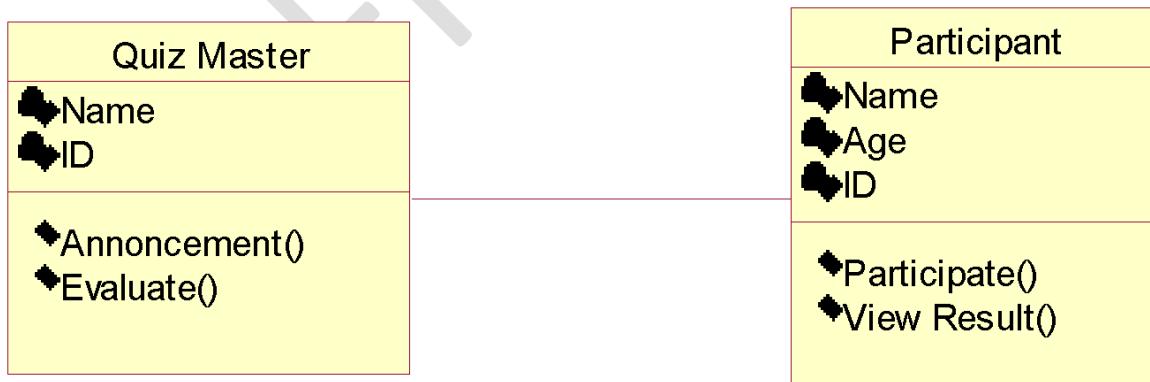
Collaboration Diagram







CLASS DIAGRAM



Forward engineering (class diagram)

CLASS DIAGRAM

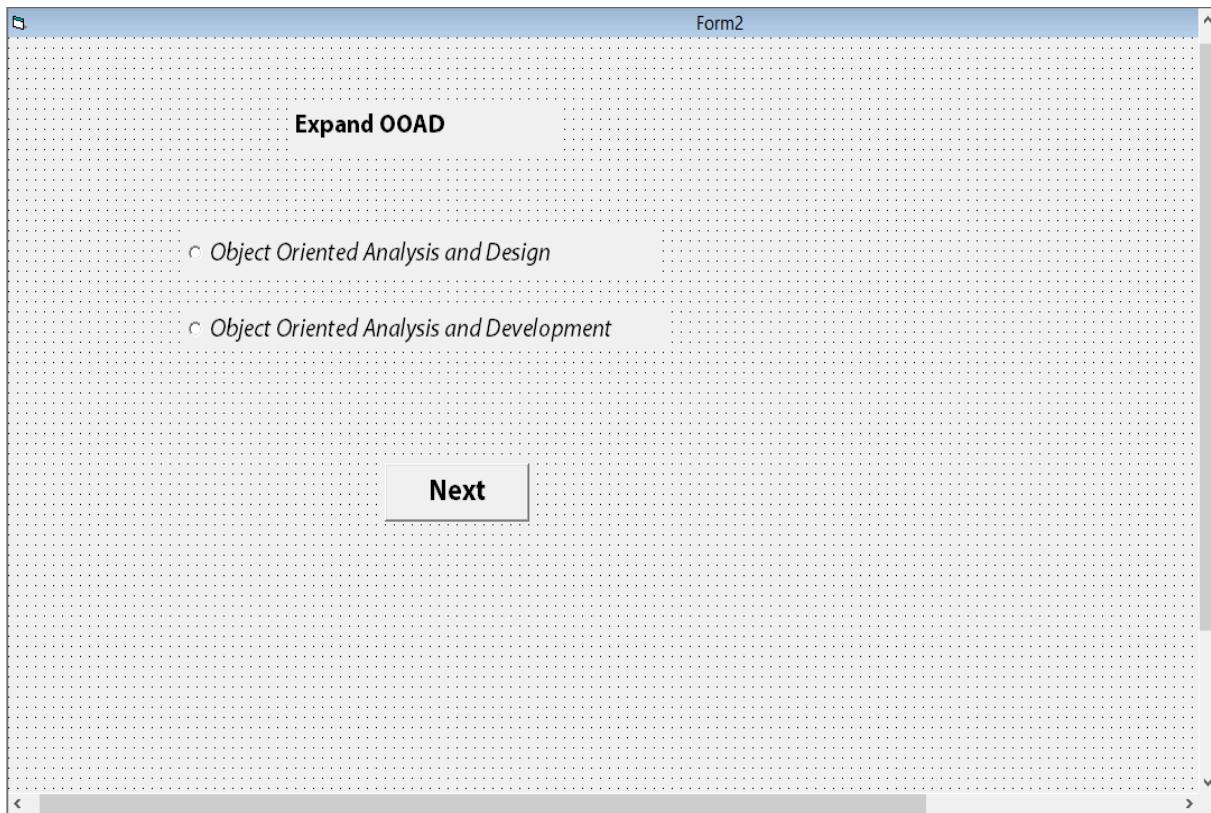


Form Design:

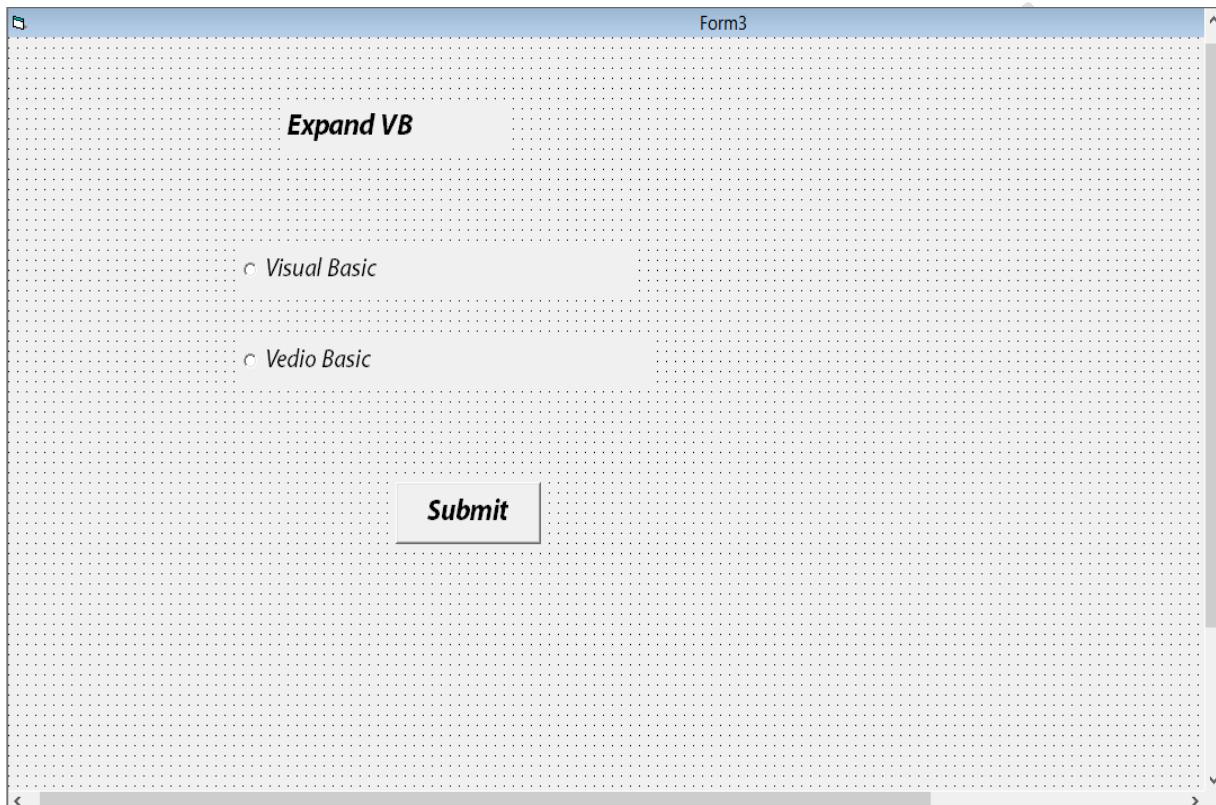
Form 1:

The screenshot shows a Windows application window titled "Form1". The title bar also displays "Quiz System". The main area of the form is titled "Quiz System". It contains four text input fields arranged vertically. To the left of each field is a label: "Name", "Age", "Address", and "Qualification". Below these four fields are two buttons: "Submit" and "Clear". The "Submit" button is positioned to the left of the "Clear" button. The entire application window is set against a background with a dotted grid pattern.

Form 2:



Form 3:



Coding:

Form 1

```
Private Sub Submit_Click()
If Val(Text2.Text) >= 18 Then
    MsgBox "Eligible Candidate. Can Proceed"
    Form2.Show
Else
    MsgBox "Ineligible Candidate"
End If
End Sub
```

```
Private Sub Clear_Click()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
End Sub
```

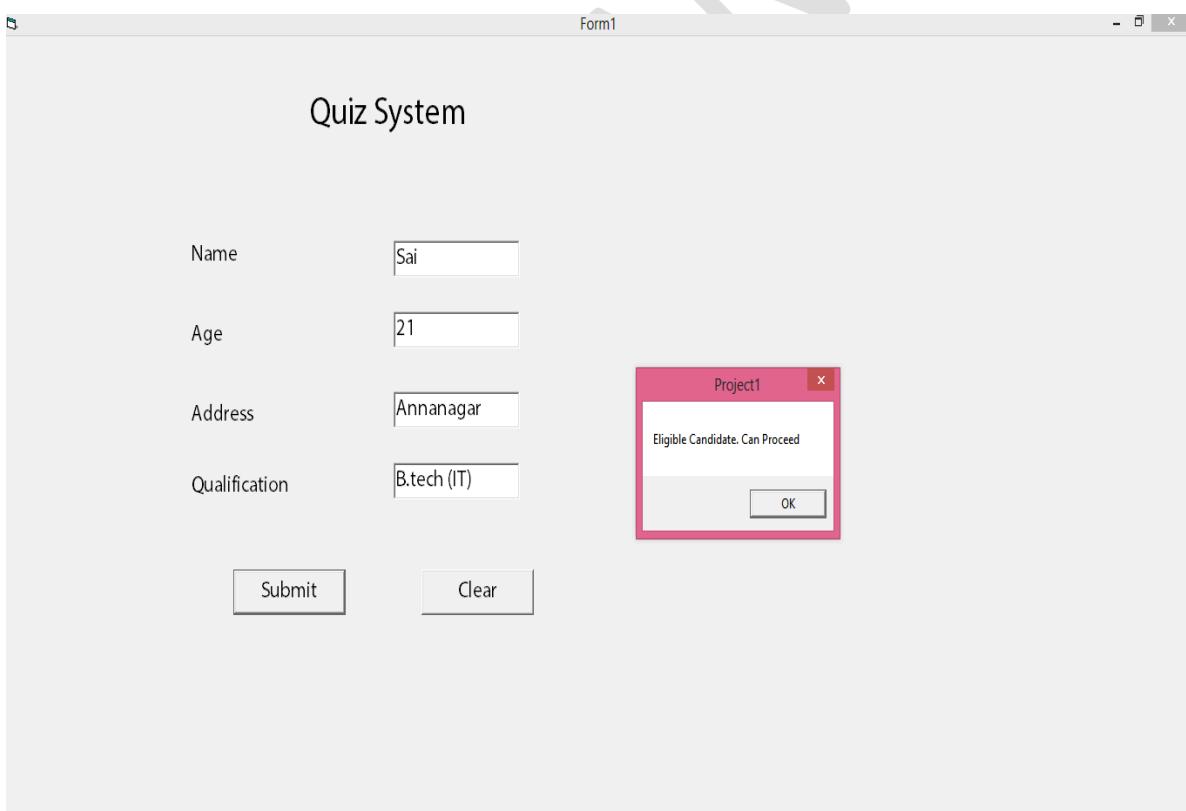
Form 2:

```
Private Sub Next_Click()
Form3.Show
End Sub
```

Form 3:

```
Private Sub Submit_Click()
If Form2.Option1.Value = True And Form3.Option1.Value = True Then
    MsgBox "Win"
Else
    MsgBox "Lost"
End If
End
End Sub
```

Output:



1 FormZ

Expand OOAD

- *Object Oriented Analysis and Design*
- *Object Oriented Analysis and Development*

Next

Result:

Thus the Quiz System project is analyzed and designed using Rational Rose and implemented in VB.

Ex.No 6.

AUTOMATING THE BANKING SYSTEM

AIM

To develop the Automating the banking System using rational rose tools, visual basic and MS access.

Problem analysis and project planning

Introduction

Banking is one of the common day to day activities of life. Nowadays it is totally different from what existed a few years back, today banking has become completely computerized, new facilities such as credit cards, debit cards & ATM has been introduced. ATM stands for Automatic Teller Machine which is basically used to withdraw money from an account.

Objectives

The objective of this software is similar to the ATM software installed in an ATM center. It first validates the pin of the ATM card, then the type of transaction is enquired and the information from the customer is validated. If it is a withdrawal the amount is asked. After the money is delivered the transaction details are updated in the database where the customer's information is stored.

Scope

The scope of the project is to design an ATM system that will help in completely automatic banking, this software is going to be designed for withdrawal and deposit and also register the transaction in the database where the customer's information is stored.

Problem Statement

ATM banking is another type of banking where the most frequent type of transaction made is withdrawal. A user may withdraw as much as he wants until his account holds a sum greater than his withdrawal amount. ATM is completely automated and there is no necessity of the ATM center being placed at the bank itself. It can be placed in the shopping malls, airports, railway stations etc.

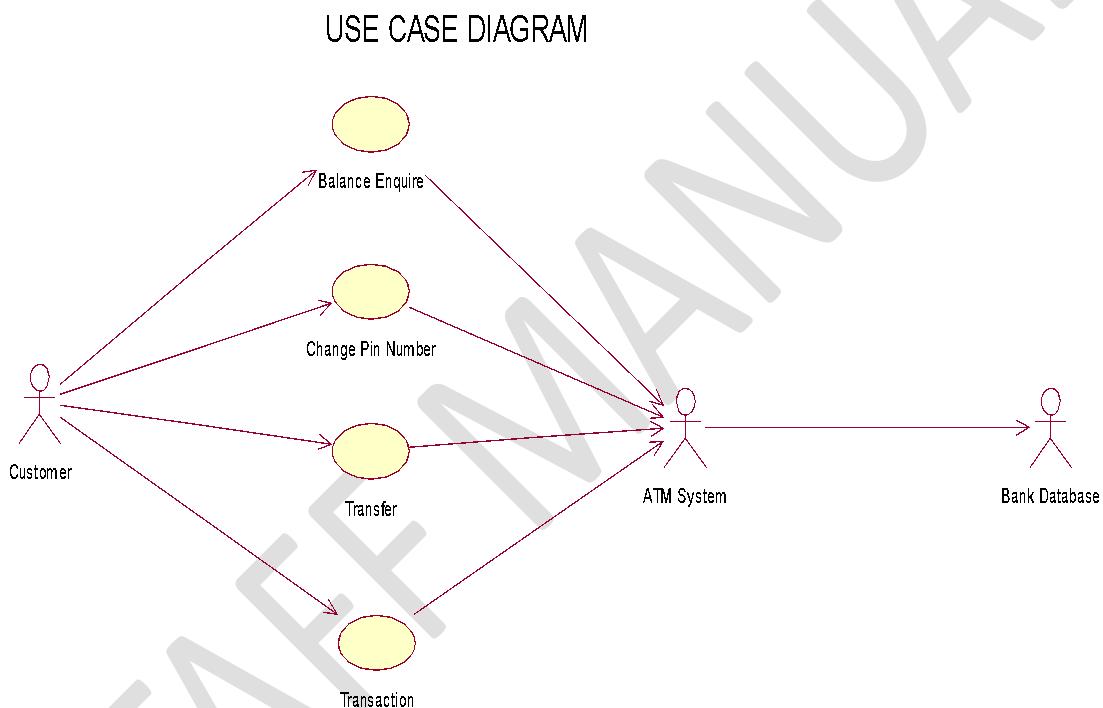
This ATM system can use any kind of interface. But it should be user friendly and not confusing. Help manuals should be provided in case any customer has problem working with the software.

The system will retain entire information of the customer who has the necessary rights to access the service. It will contain the balance amount in the account, rate of interest, any special allowance for that customer and most of all pin number of the customer. The ATM system should be compatible with any kind of database such as MS-ACCESS, DB2, ORACLE, SQL, SERVER etc. the emphasis here is on consistency.

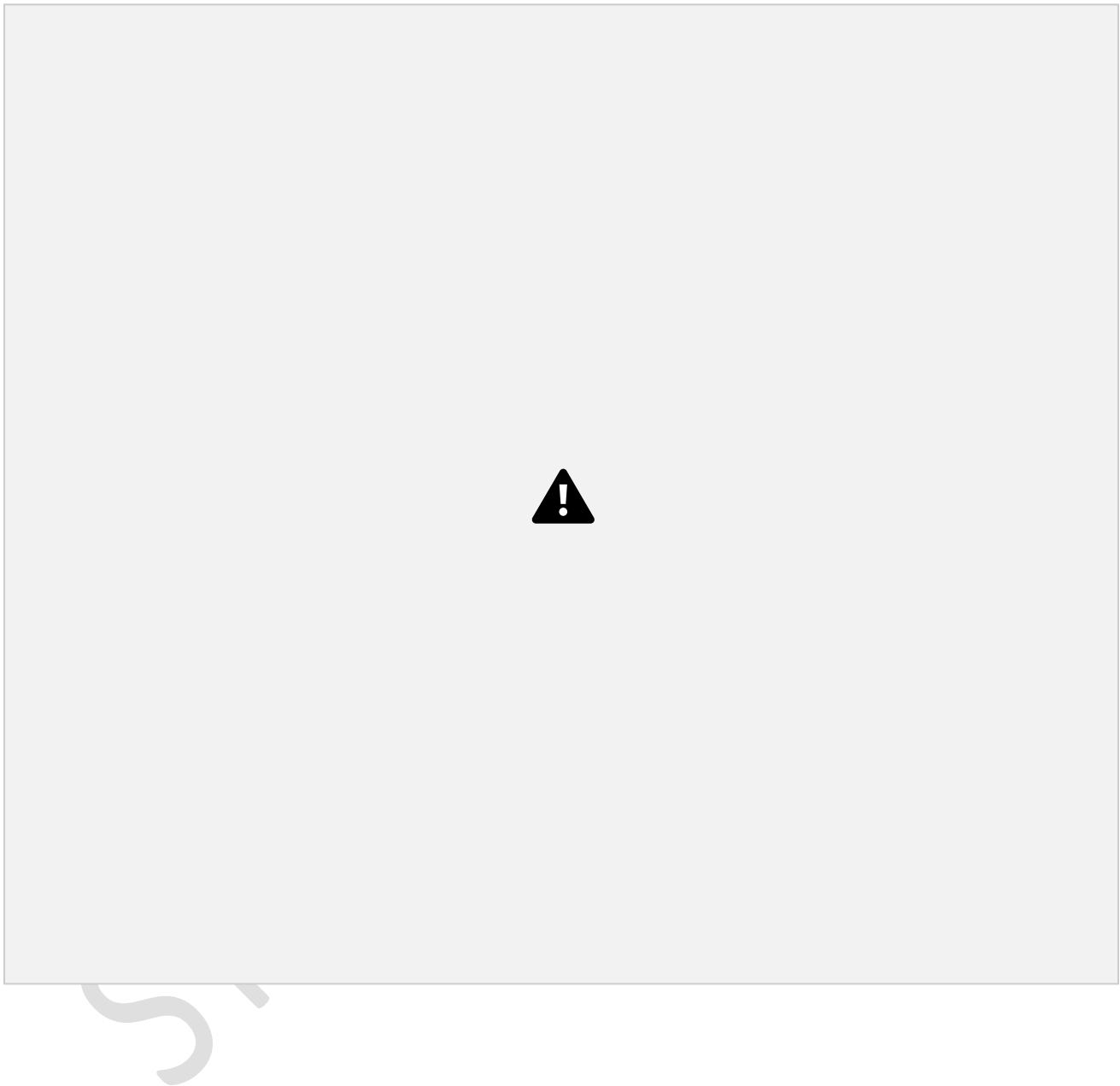
Some customers may have availed some special offers on their ATM cards. So this must be taken care of and the appropriate data should be dealt with.

The ATM should provide easy access to the data for the customer. It should also have a highly secure interface so that one can take money on behalf of others. So security is one of the paramount aspects in ATM.

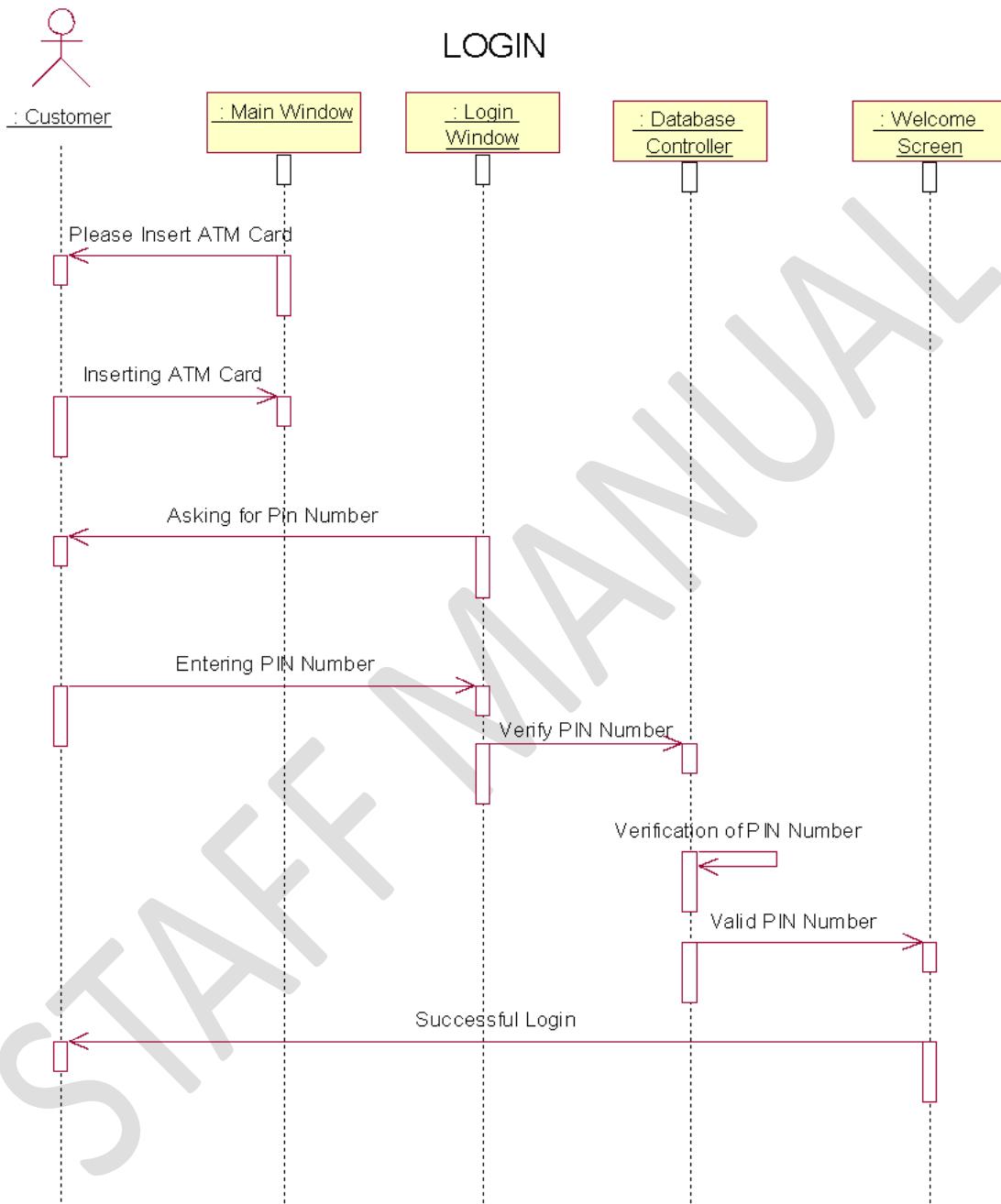
UML Diagrams:

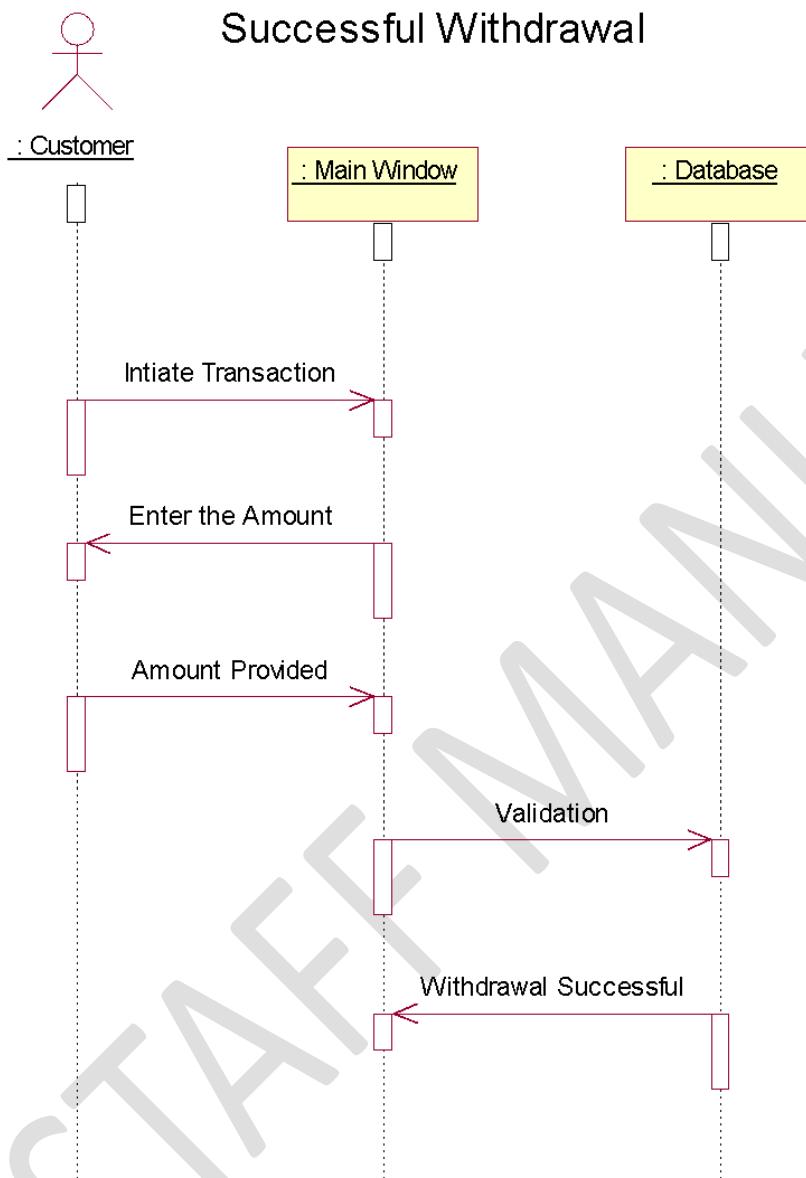


Activity Diagram:

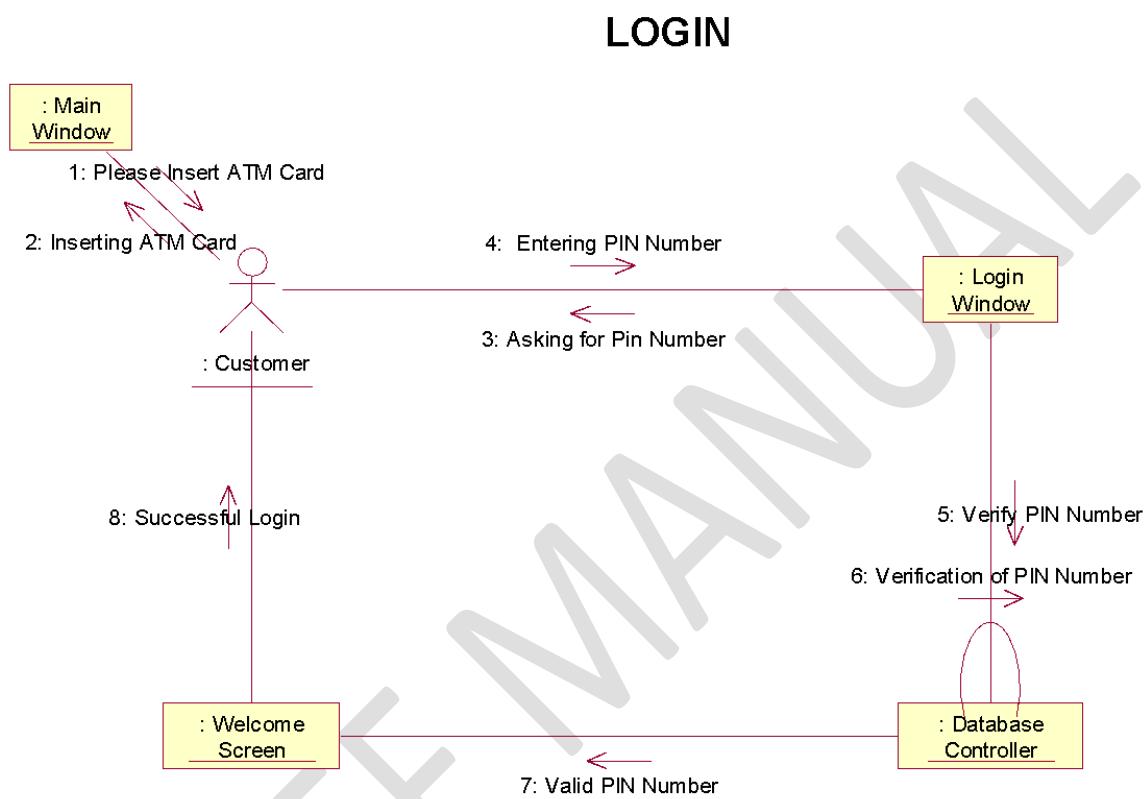


Sequence Diagrams:

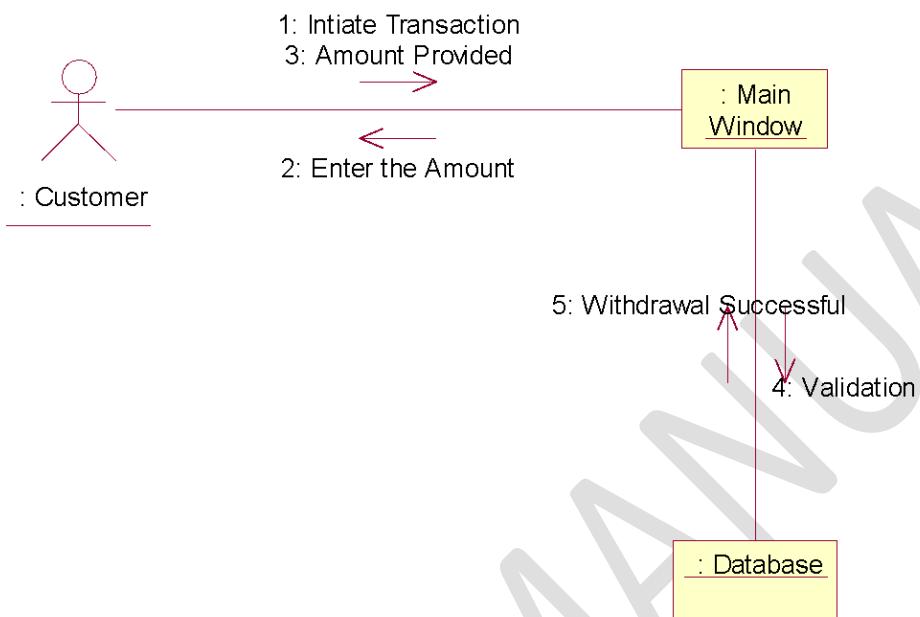




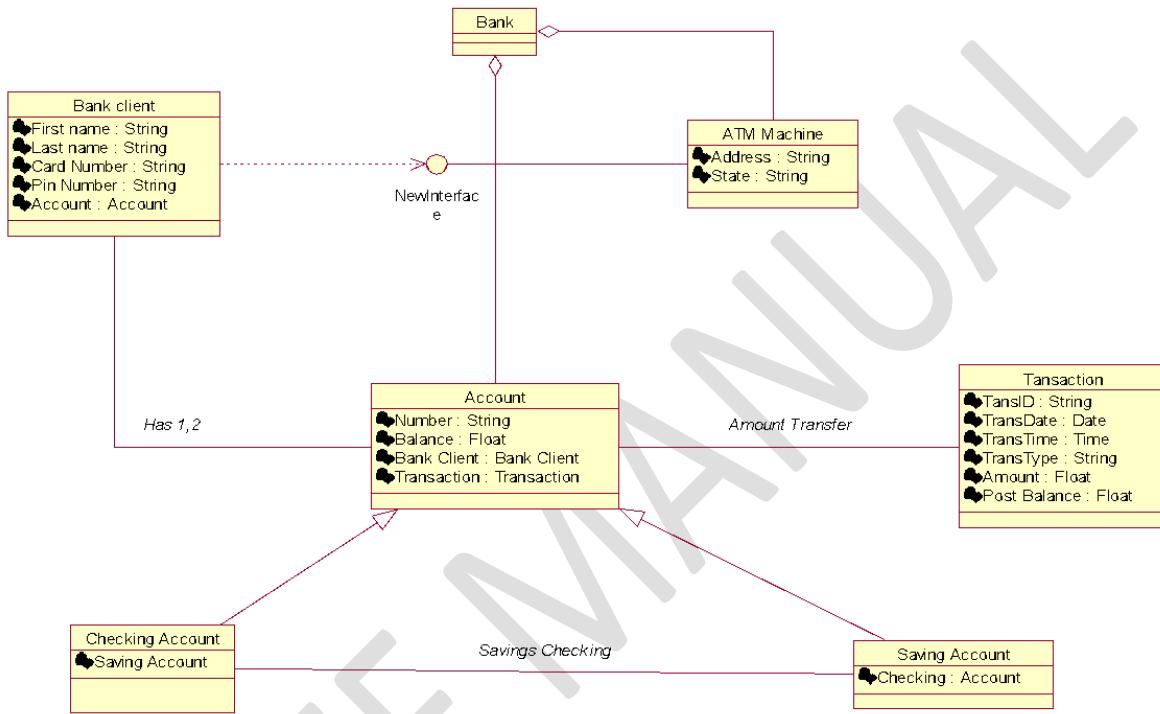
Collaboration Diagram



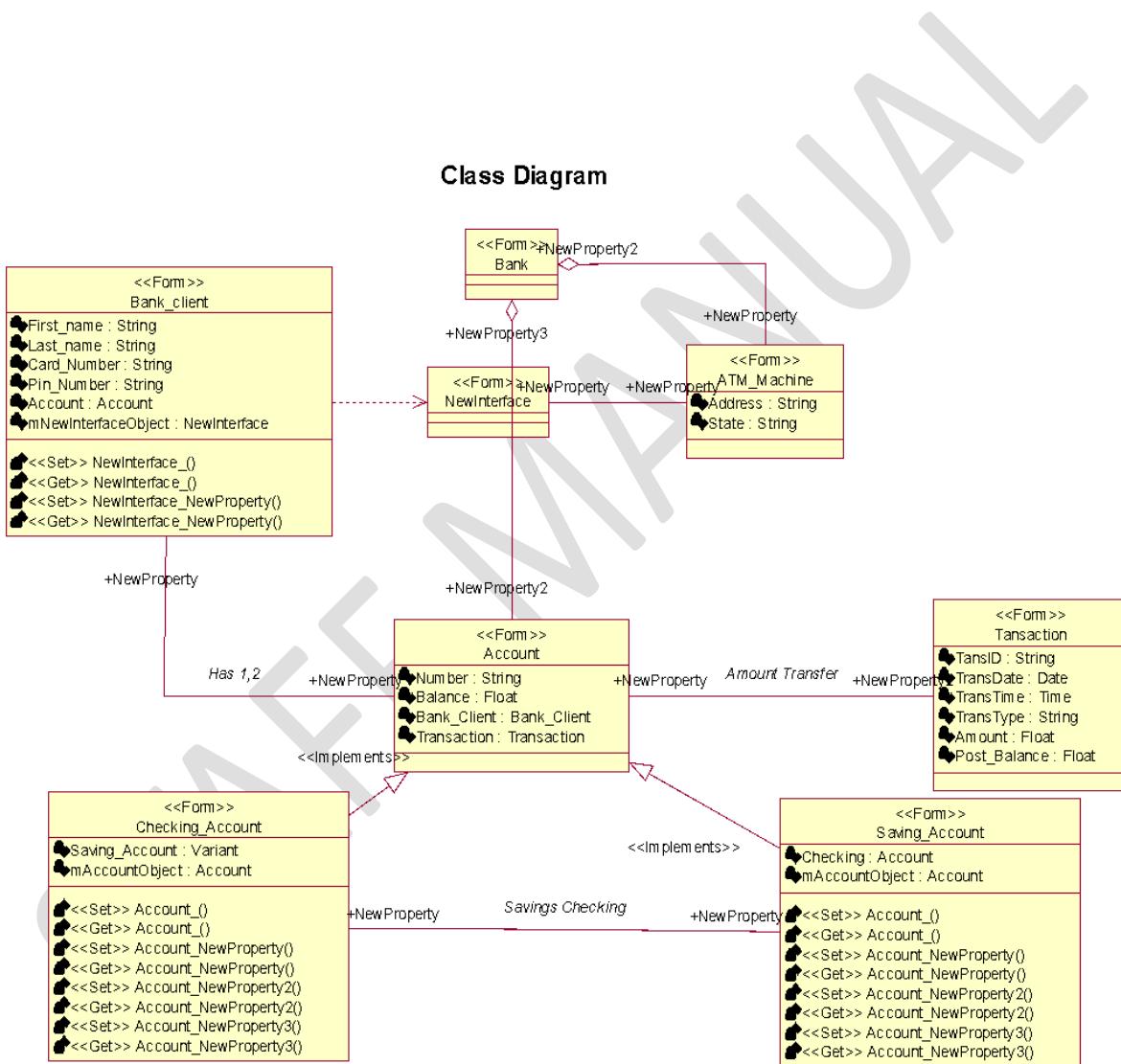
Successful Withdrawal



Class Diagram

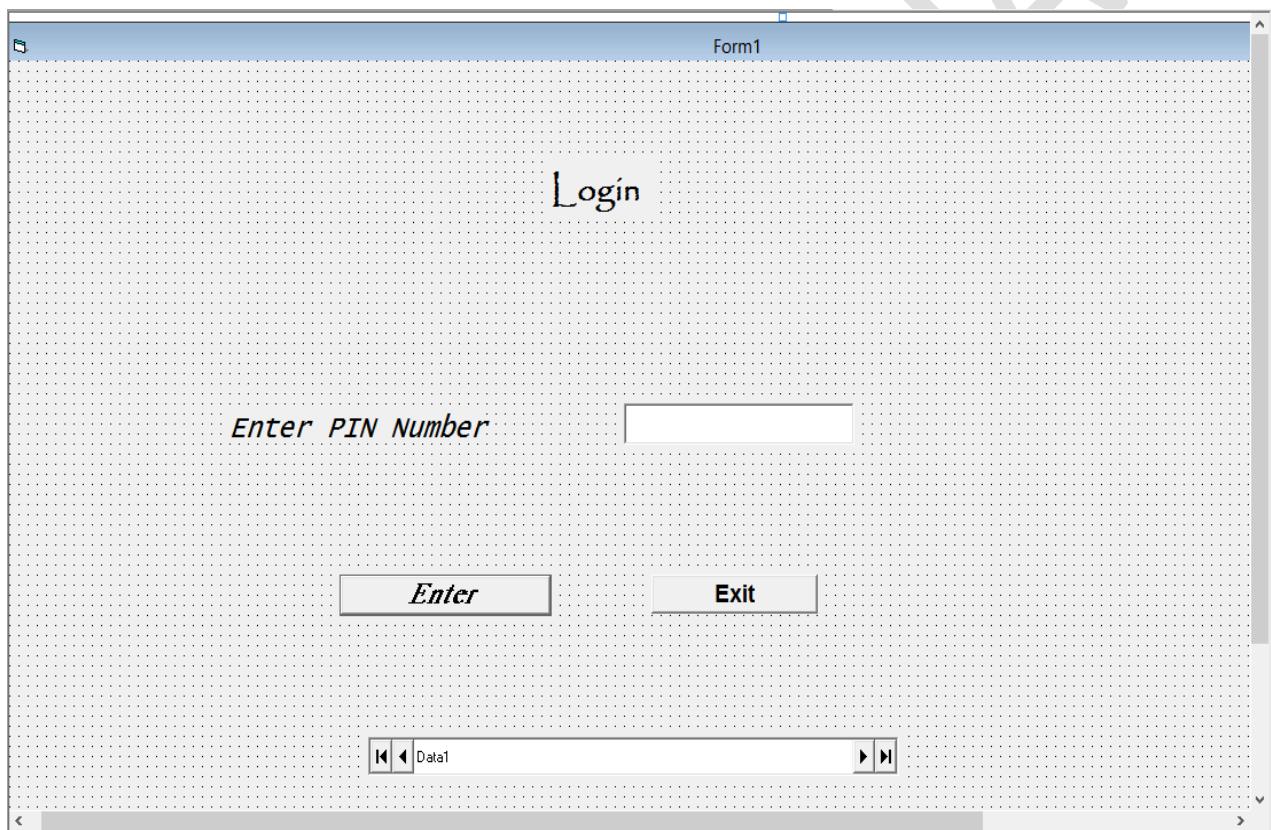


Forward Engineering Class Diagram

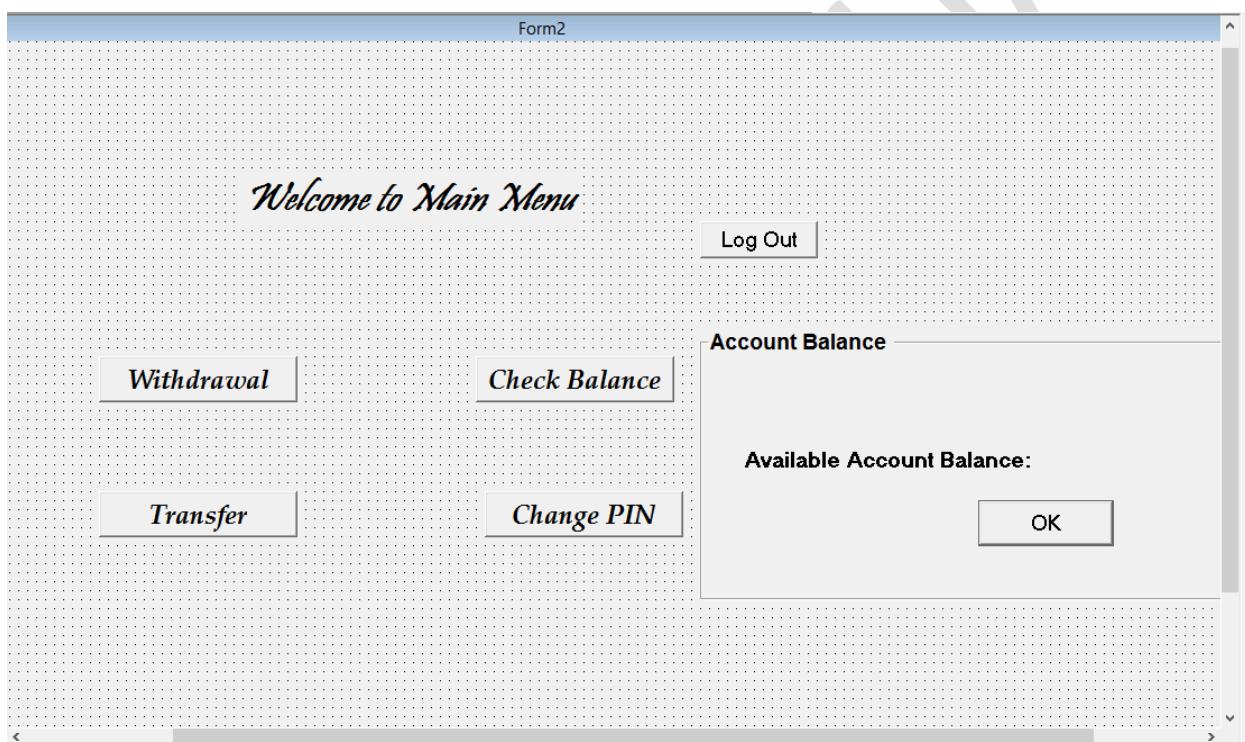


From Design

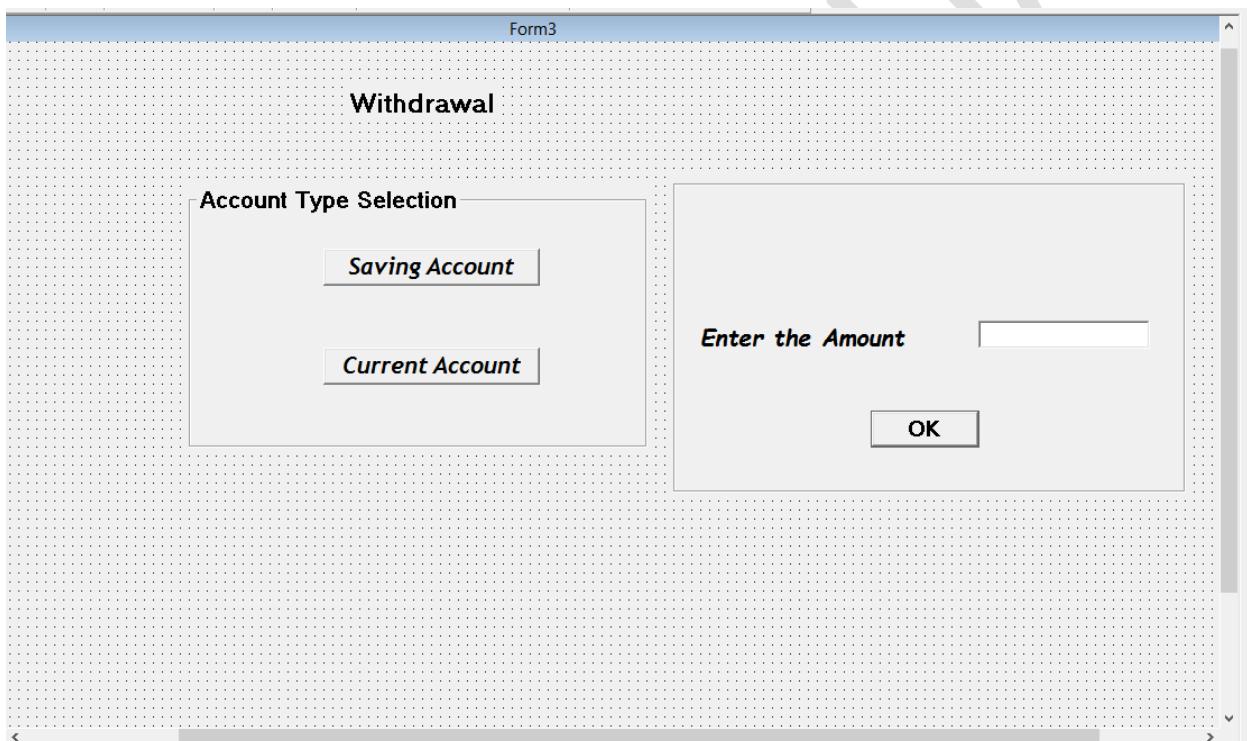
Form 1:



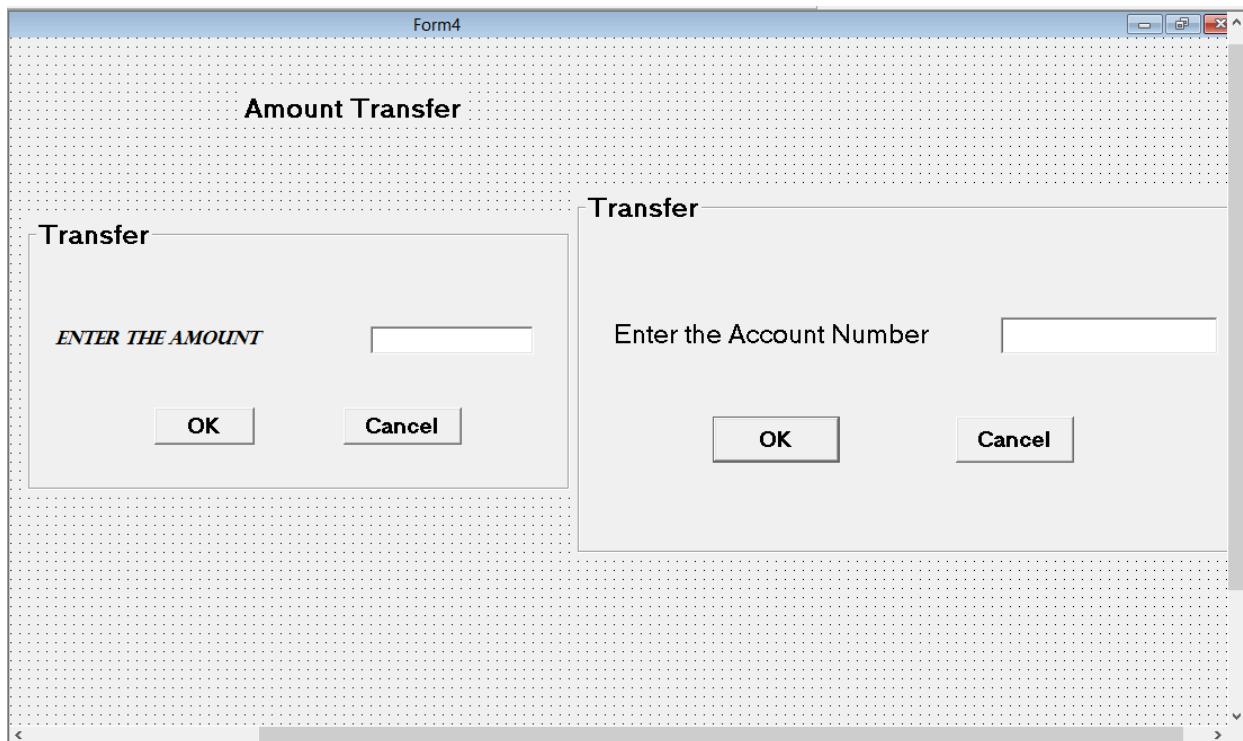
Form 2:



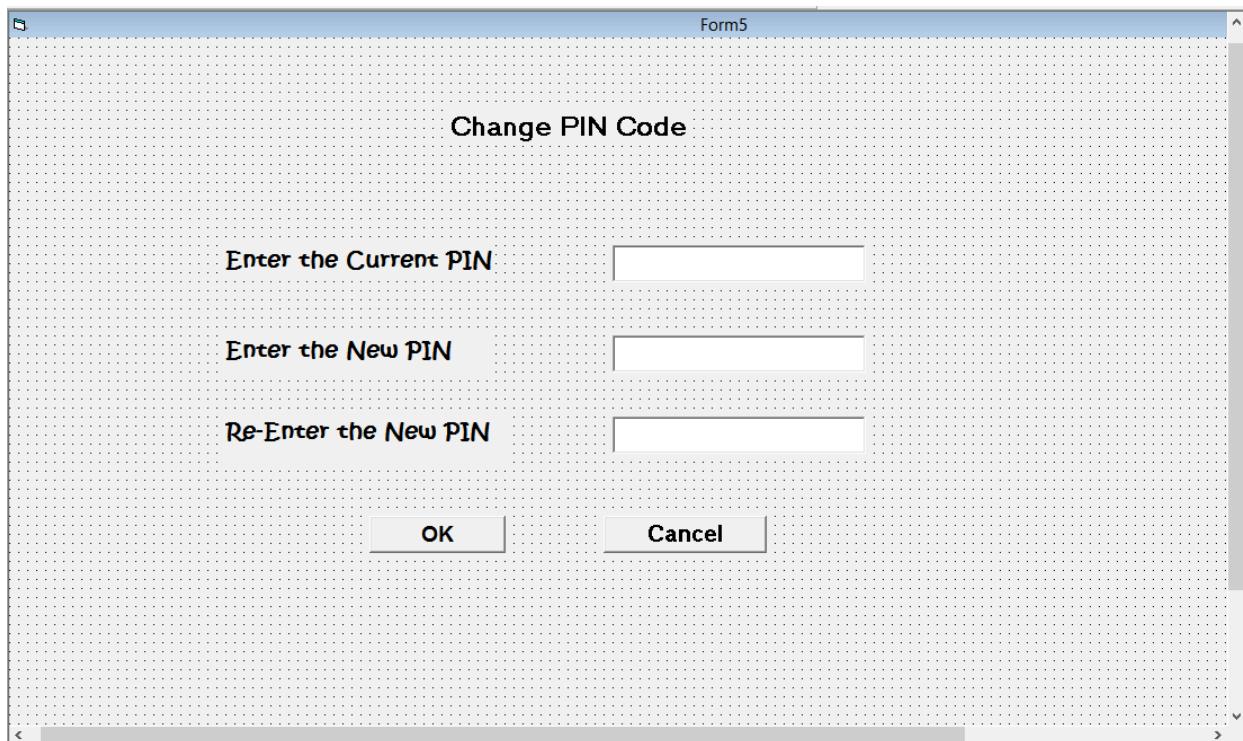
Form 3:



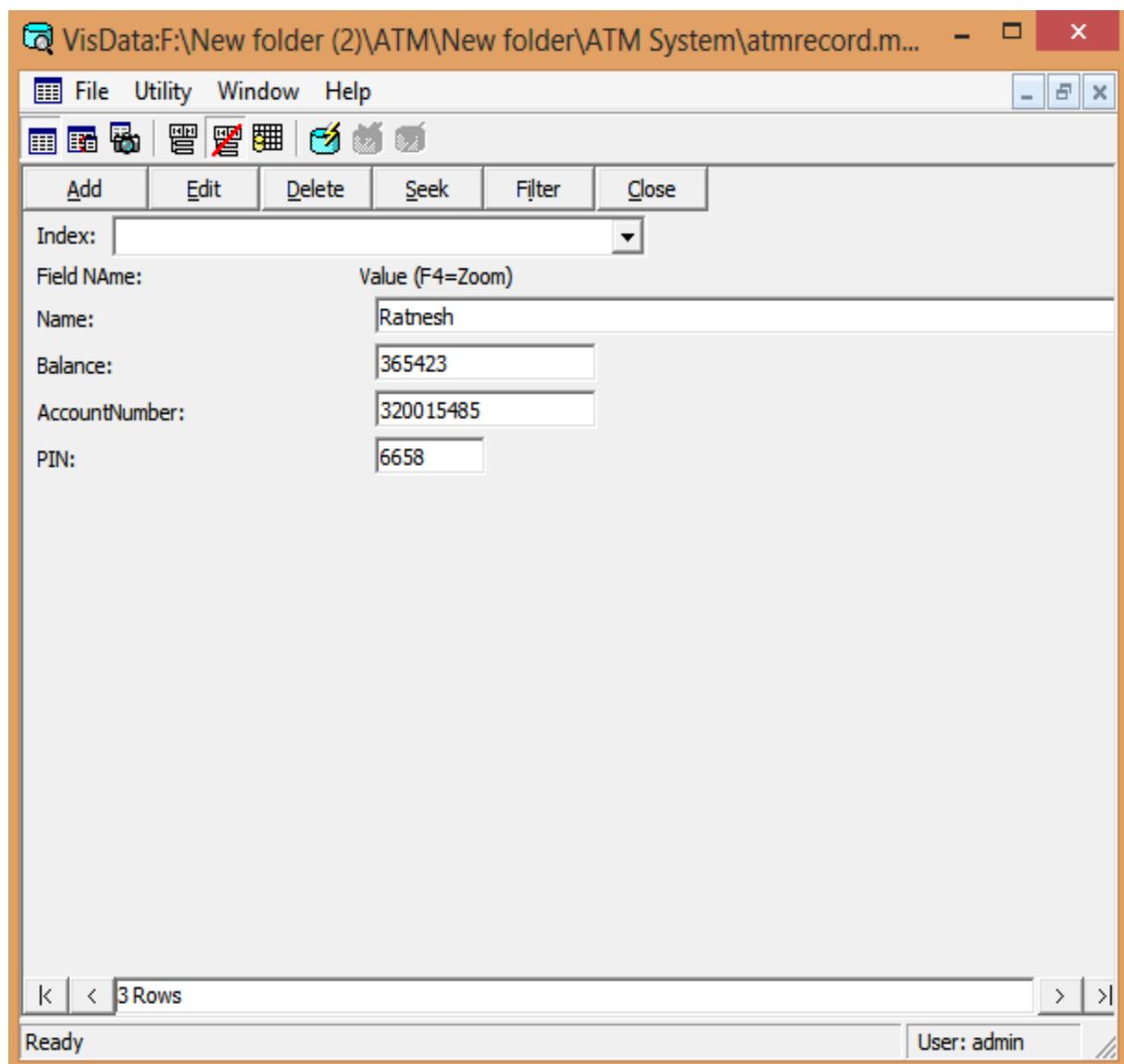
Form 4:



Form 5:



DATABASE DESIGN:



Coding:

Form 1:

Public bookmarkvar As Variant

Private Sub Command1_Click()

End

End Sub

Private Sub Enter_Click()

With Data1

.Recordset.MoveFirst

Do Until .Recordset.EOF

If (Val(Text1.Text) = .Recordset.Fields("PIN")) Then

bookmarkvar = .Recordset.Bookmark

Form2.Show

Unload Me

Exit Sub

Else

.Recordset.MoveNext

End If

Loop

MsgBox ("INCORRECT PIN ! Retry")

Text1.Text = ""

Text1.SetFocus

End With

End Sub

Form 2:

Private Sub Command1_Click()

Unload Me

Form3.Show

End Sub

Private Sub Command2_Click()

Frame1.Visible = True

Form1.Data1.Recordset.Bookmark = Form1.bookmarkvar

Label3.Caption = Form1.Data1.Recordset.Fields("Balance")

End Sub

Private Sub Command3_Click()

Unload Me

Form4.Show

End Sub

Private Sub Command4_Click()

Unload Me

Form5.Show

End Sub

Private Sub Command5_Click()

Frame1.Visible = False

End Sub

Private Sub Command6_Click()

Form1.Show

Unload Me

End Sub

Private Sub Form_Load()

Frame1.Visible = False

Label3.Caption = ""

End Sub

Form 3:

Private Sub Command1_Click()

Frame1.Visible = False

Frame2.Visible = True

End Sub

Private Sub Command2_Click()

Frame1.Visible = False

Frame2.Visible = True

End Sub

Private Sub Command3_Click()

Form1.Data1.Recordset.Bookmark = Form1.bookmarkvar

With Form1.Data1 .Recordset.Edit

If Val(Text1.Text) < .Recordset.Fields("Balance") Then

.Recordset.Fields("Balance") = .Recordset.Fields("Balance") - Val(Text1.Text)

MsgBox "Transaction Completed "

.Recordset.Update

Form2.Show

```
Unload Me
Exit Sub
Else
MsgBox "Insufficient Balance"
End If
End With
End Sub
Private Sub Form_Load()
Frame2.Visible = False
Frame1.Visible = True
End Sub
Form 4:
Dim a As Currency
Private Sub Command1_Click()
Frame1.Visible = True
Frame2.Visible = False
End Sub
Private Sub Command2_Click()
Form2.Show
End Sub
Private Sub Command3_Click()
Form1.Data1.Recordset.Bookmark = Form1.bookmarkvar
With Form1.Data1
a = .Recordset.Fields("AccountNumber")
.Recordset.MoveFirst
End With
End Sub
```

```
Do Until .Recordset.EOF

If (Val(Text1.Text) = .Recordset.Fields("AccountNumber")) Then

    .Recordset.Edit

    .Recordset.Fields("Balance") = .Recordset.Fields("Balance") + Val(Text2.Text)

    .Recordset.Update

    .Recordset.MoveFirst

    Do Until .Recordset.EOF

        .Recordset.Edit

        If (.Recordset.Fields("AccountNumber") = a) Then

            If Val(Text2.Text) < .Recordset.Fields("Balance") Then

                .Recordset.Fields("Balance") = .Recordset.Fields("Balance") - Val(Text2.Text)

            MsgBox "Transaction Completed"

            .Recordset.Update

            Form2.Show

            Unload Me

            Exit Sub

        Else

            MsgBox "Insufficient Balance"

            Exit Sub

        End If

    Else

        .Recordset.MoveNext

    End If

Loop

Else
```

```
.Recordset.MoveNext  
End If  
Loop  
MsgBox ("INCORRECT Account Number ! Retry")
```

```
Text1.Text = ""
```

```
Text1.SetFocus
```

```
End With
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Form2.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Frame2.Visible = True
```

```
Frame1.Visible = False
```

```
End Sub
```

```
Private Sub Frame2_DragDrop(Source As Control, X As Single, Y As Single)
```

```
End Sub
```

Form 5:

```
Private Sub Command1_Click()
```

```
Form1.Data1.Recordset.Bookmark = Form1.bookmarkvar
```

```
With Form1.Data1.Recordset
```

```
If Val(Text1.Text) = .Fields("PIN") Then
```

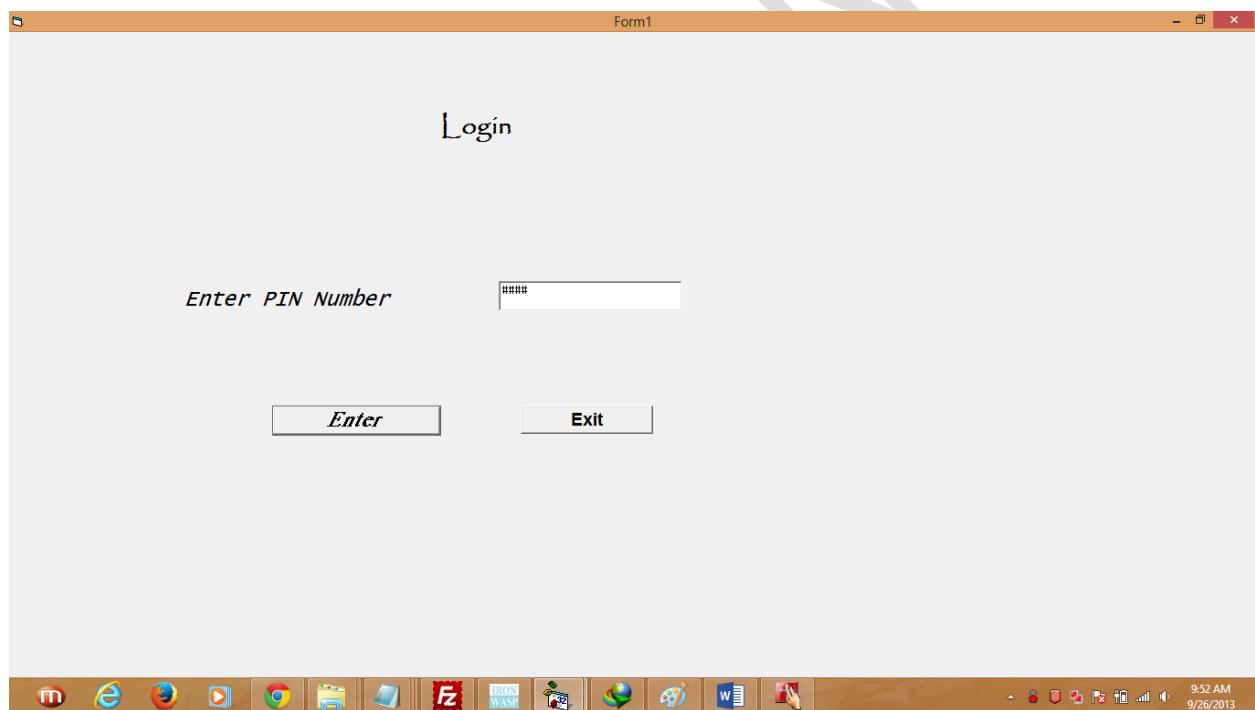
```
    If Val(Text2.Text) = Val(Text3.Text) Then
```

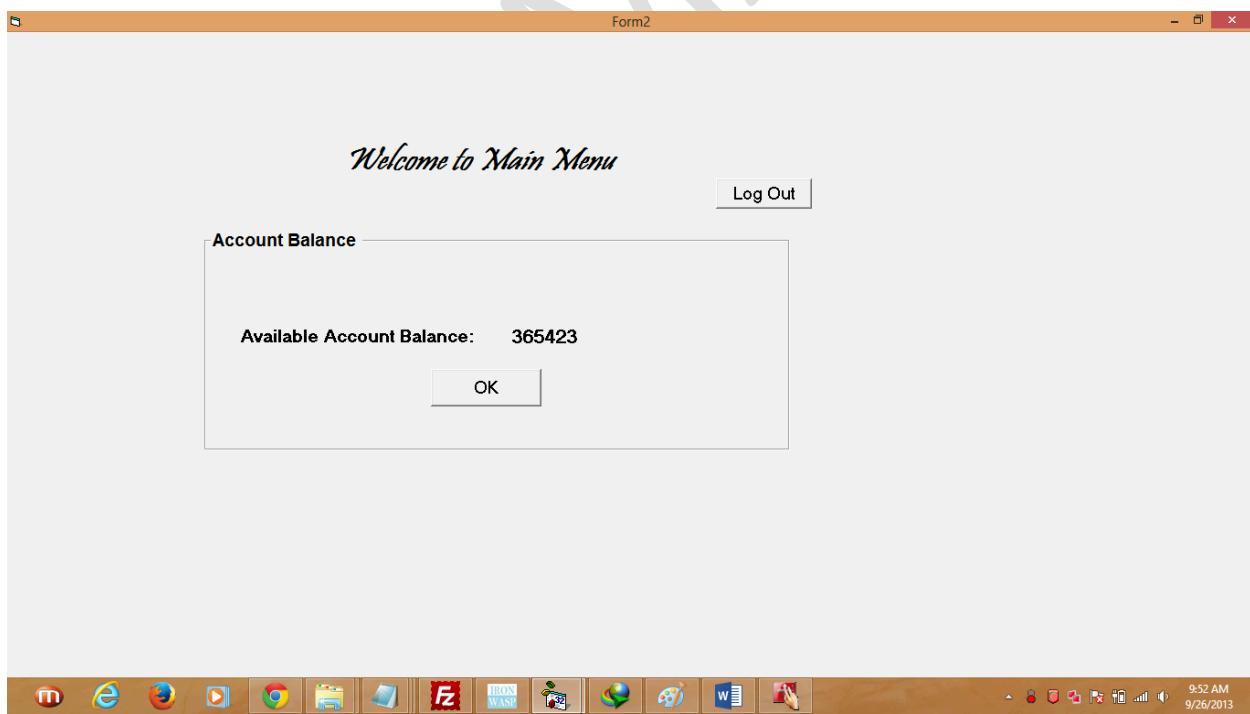
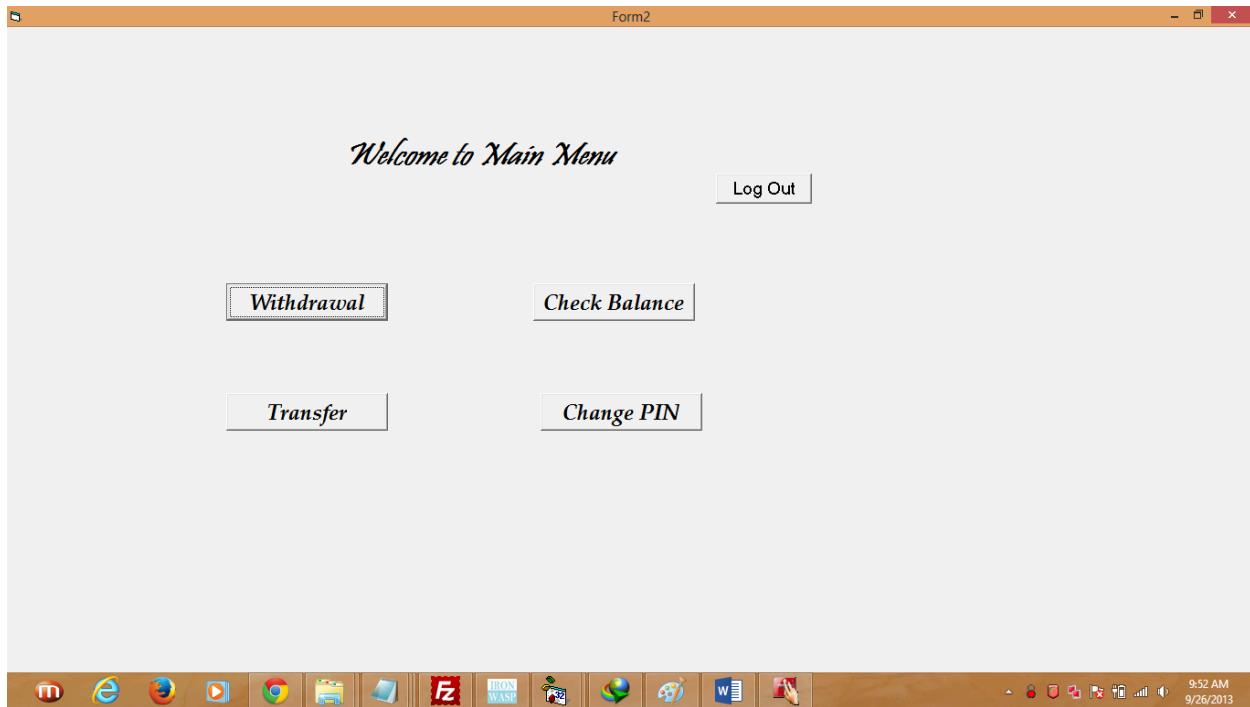
```
        .Edit
```

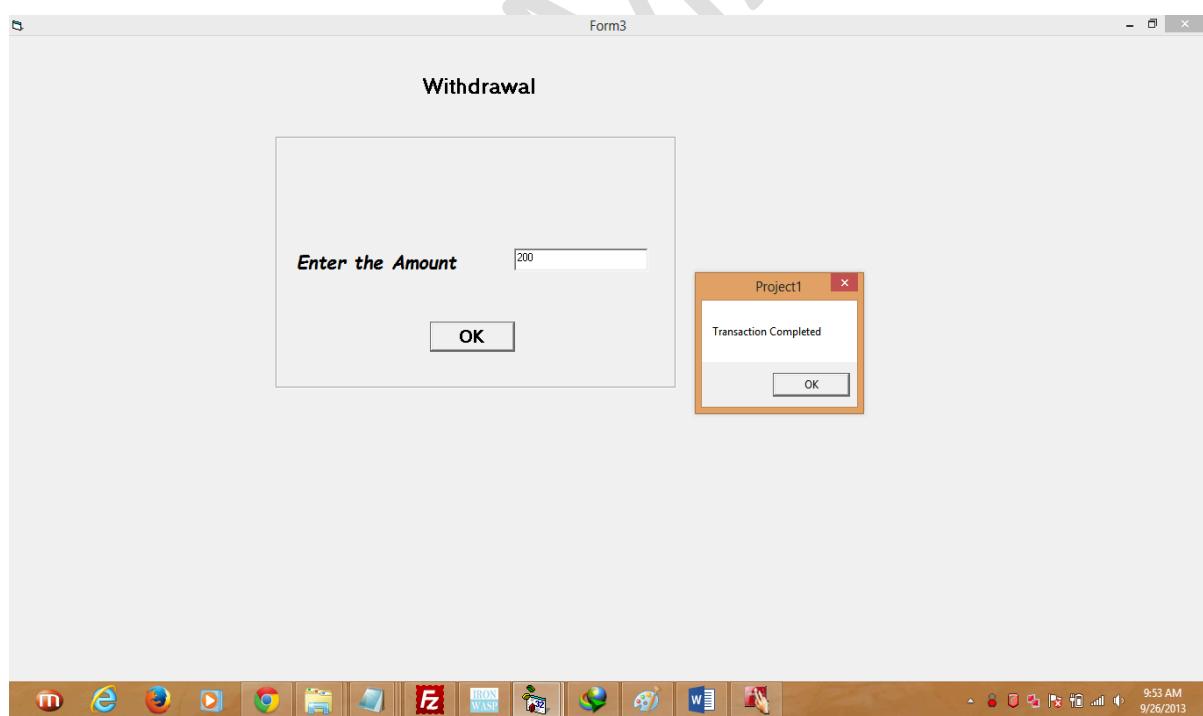
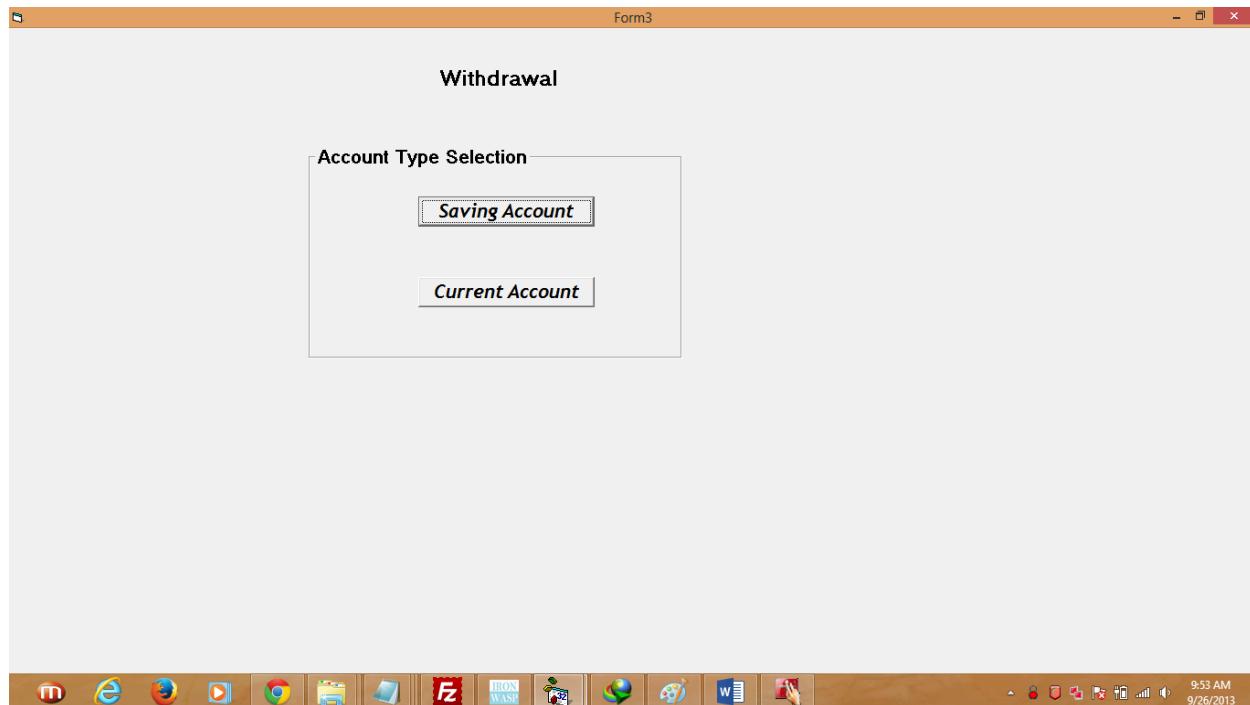
```
        .Fields("PIN") = Val(Text2.Text)
```

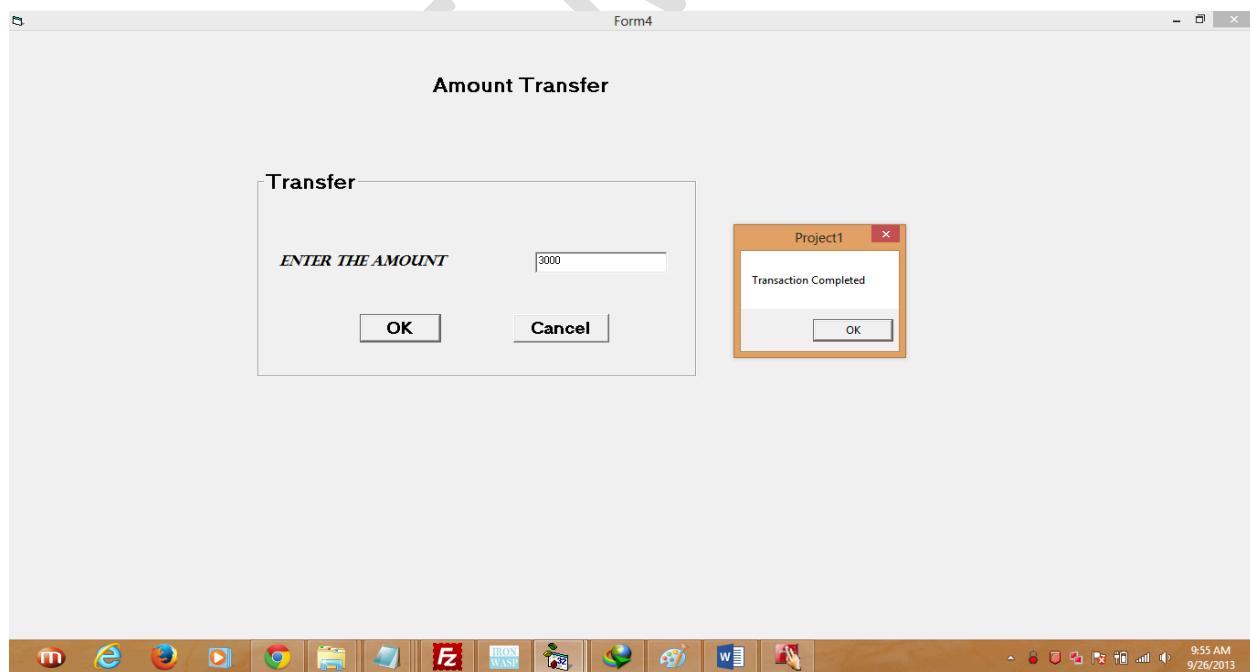
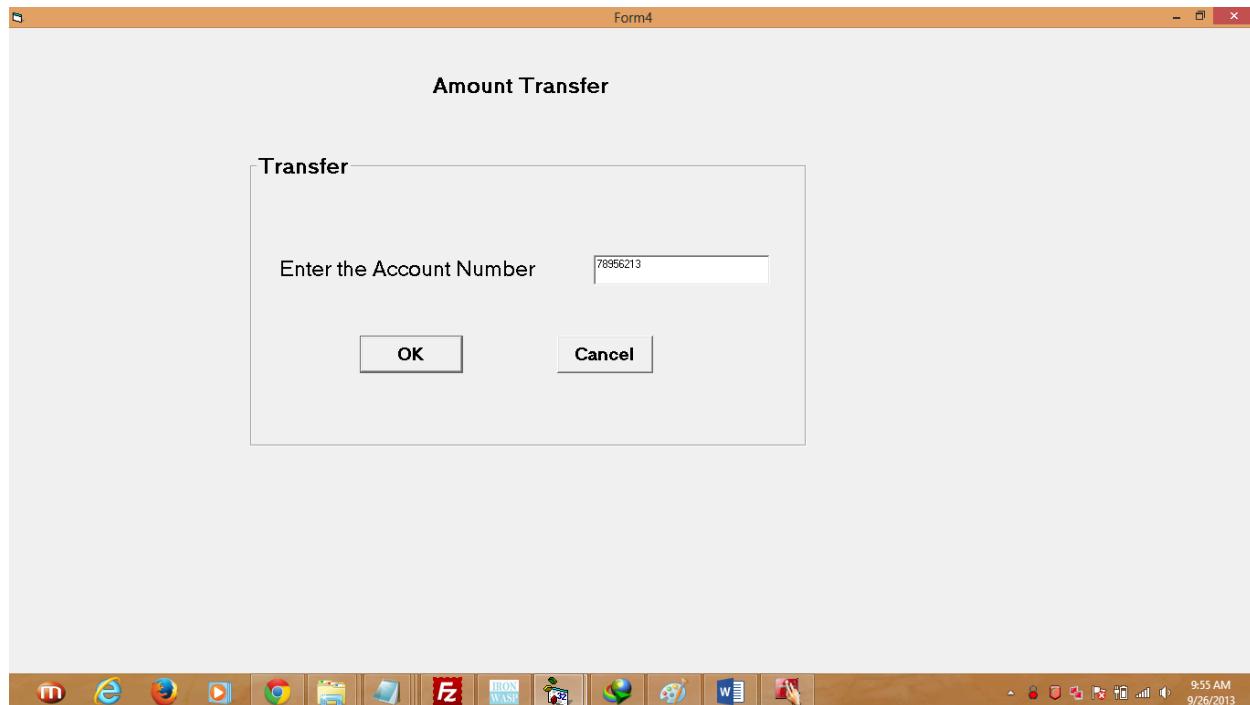
```
.Update  
MsgBox ("PIN CHANGED ")  
Form2.Show  
Unload Me  
Exit Sub  
Else  
 MsgBox ("NEW PIN DIDN'T MATCHED")  
 Exit Sub  
End If  
Exit Sub  
Else  
 MsgBox ("INVALID CURRENT PIN")  
 Exit Sub  
End If  
End With  
End Sub  
  
Private Sub Command2_Click()  
Form2.Show  
Unload Me  
End Sub  
  
Private Sub Form_Load()  
End Sub
```

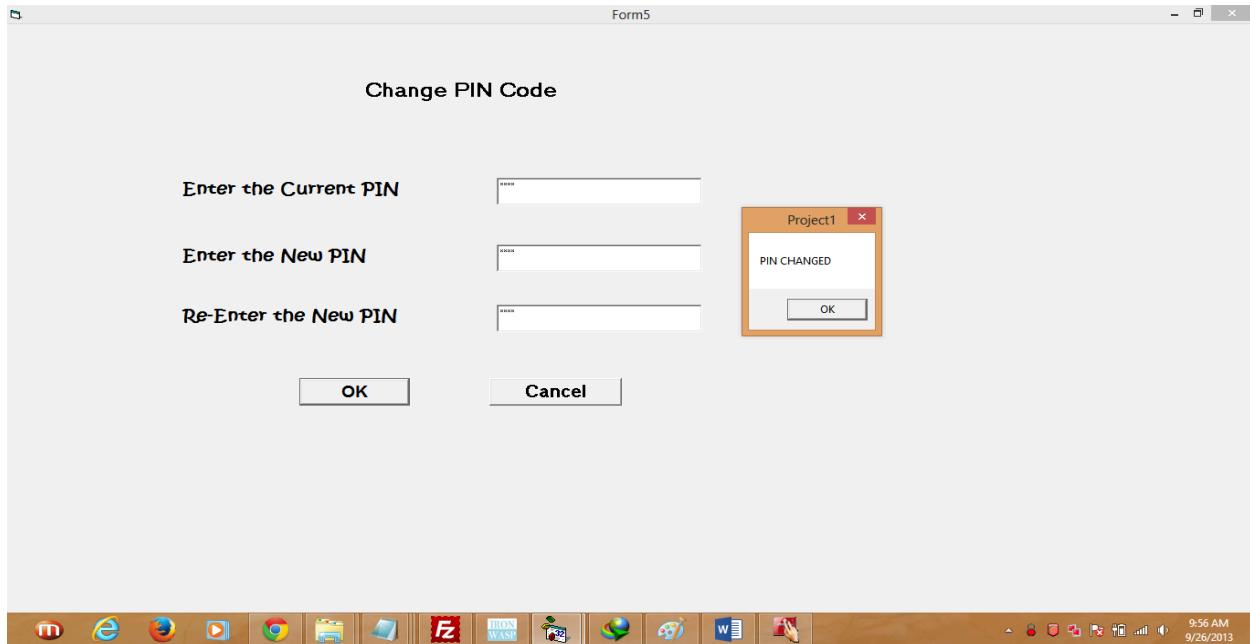
OUTPUT











Result:

Thus the ATM System project is analyzed, Design using Rational rose implemented VB.

EX.NO:7

COURSE REGISTRATION SYSTEM

AIM

To develop the Course Registration System using rational rose tools, visual basic and MS access.

1. Problem analysis and project planning

Introduction

This software is designed in such a way that it receives the name and other particulars from the student. Based on marks the student has scored the list of possible branches that will accommodate for the student will be displayed. Only work for the student is what he has to fill in the form and submit it.

Objectives

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering himself. This software will reduce the paper work. This also reduces the time delay.

Scope

The student is first requested to fill the form. This form will contain important particulars of the student like his name, DOB, preferred branch, his marks. Once the student fills it, a unique id number will be generated. An important thing within this is to decide what mode the student opts to make the payment. It may either be by demand draft or credit card. As soon as a student is registered then the number of seats available is displayed.

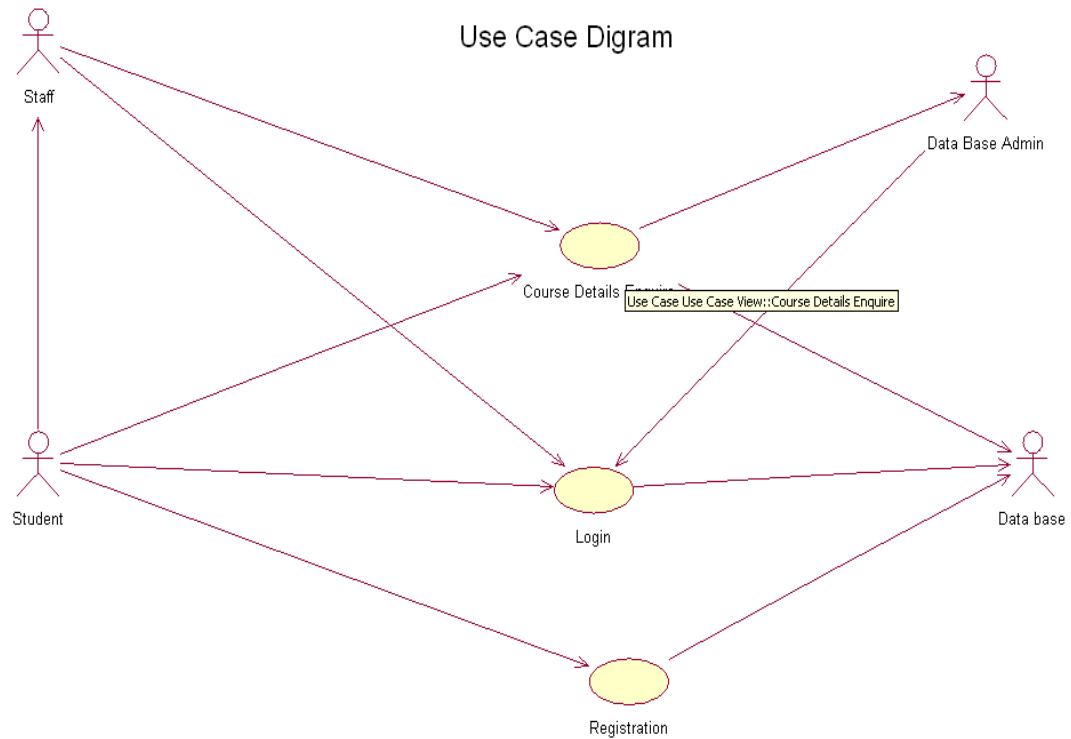
Problem Statement

As project developers we developed a new course registration system to replace the existing manual registration since manual system are prone to errors and take more time.

The system made is user friendly and reduces the burden of users. Our system can be made available even in the website of a college.

Students can easily register the course in our system without any difficulty and can easily understand and also the time taken for registration is less when compared to manual registration. Options are given to the student to select their electives and also it shows the number of papers available along with the number of students who have registered and also the number of days for particular elective per semester is also displayed at the side. This makes their work easier as compared to manual registration since they need to make a copy of HOD, staff separately and even if one is missed out the whole process has to be redone. Their information will be stored as soon as they are registered.

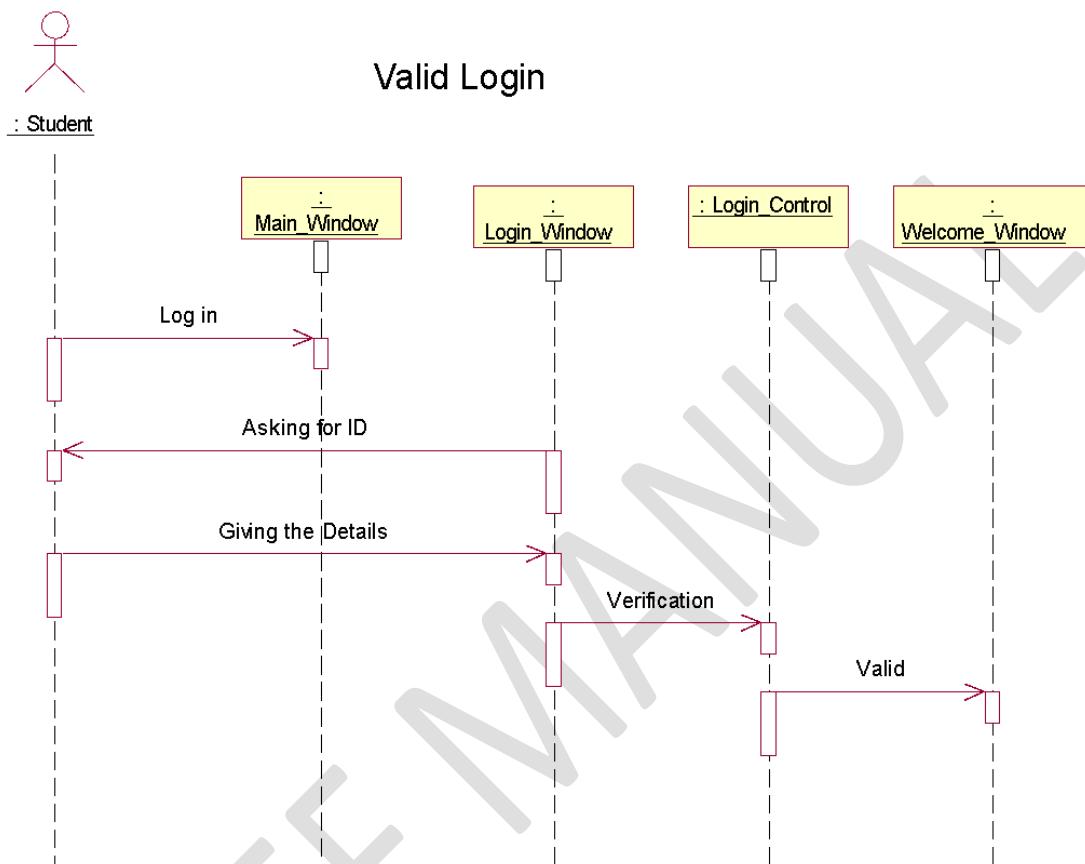
UML Diagrams:

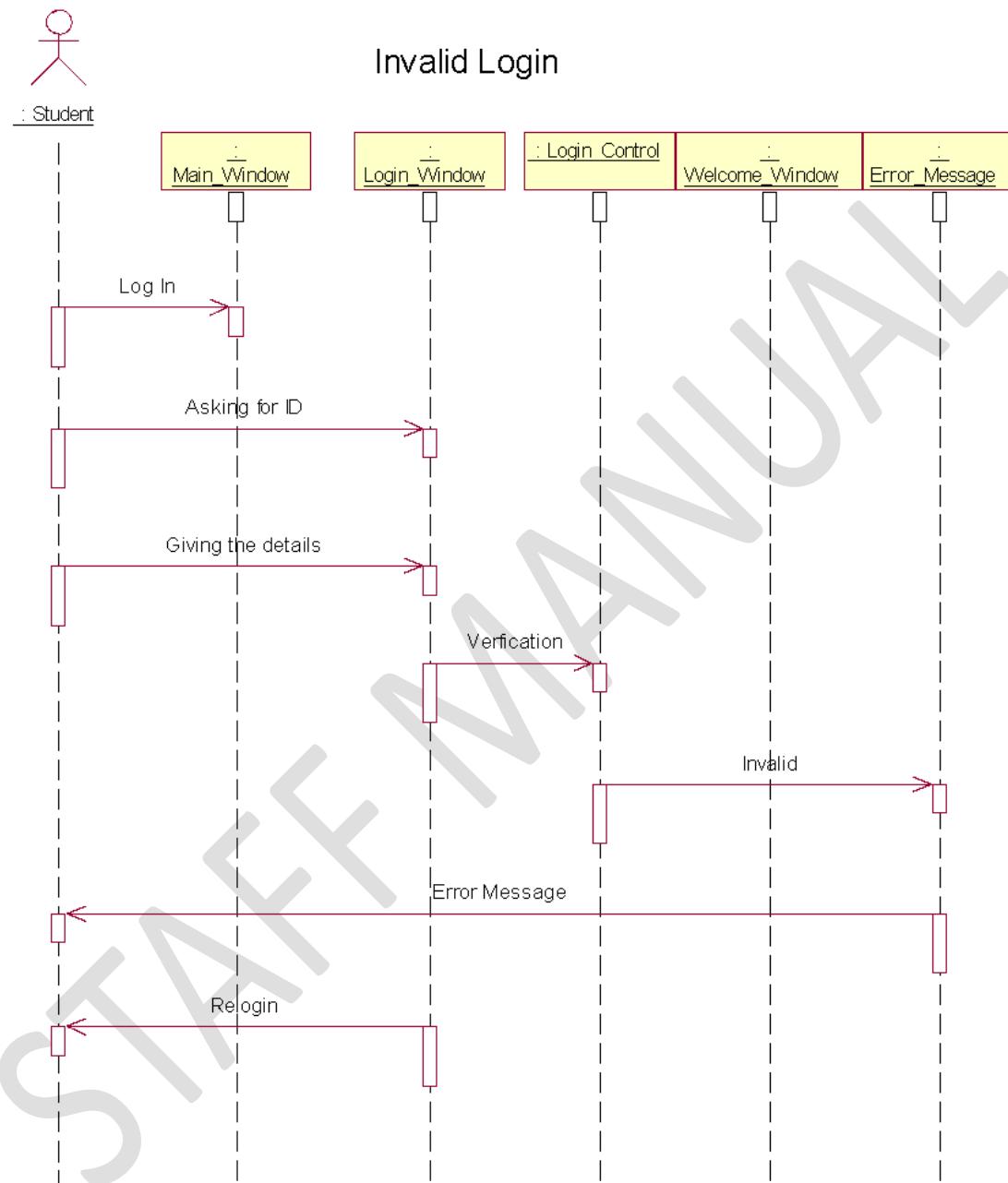


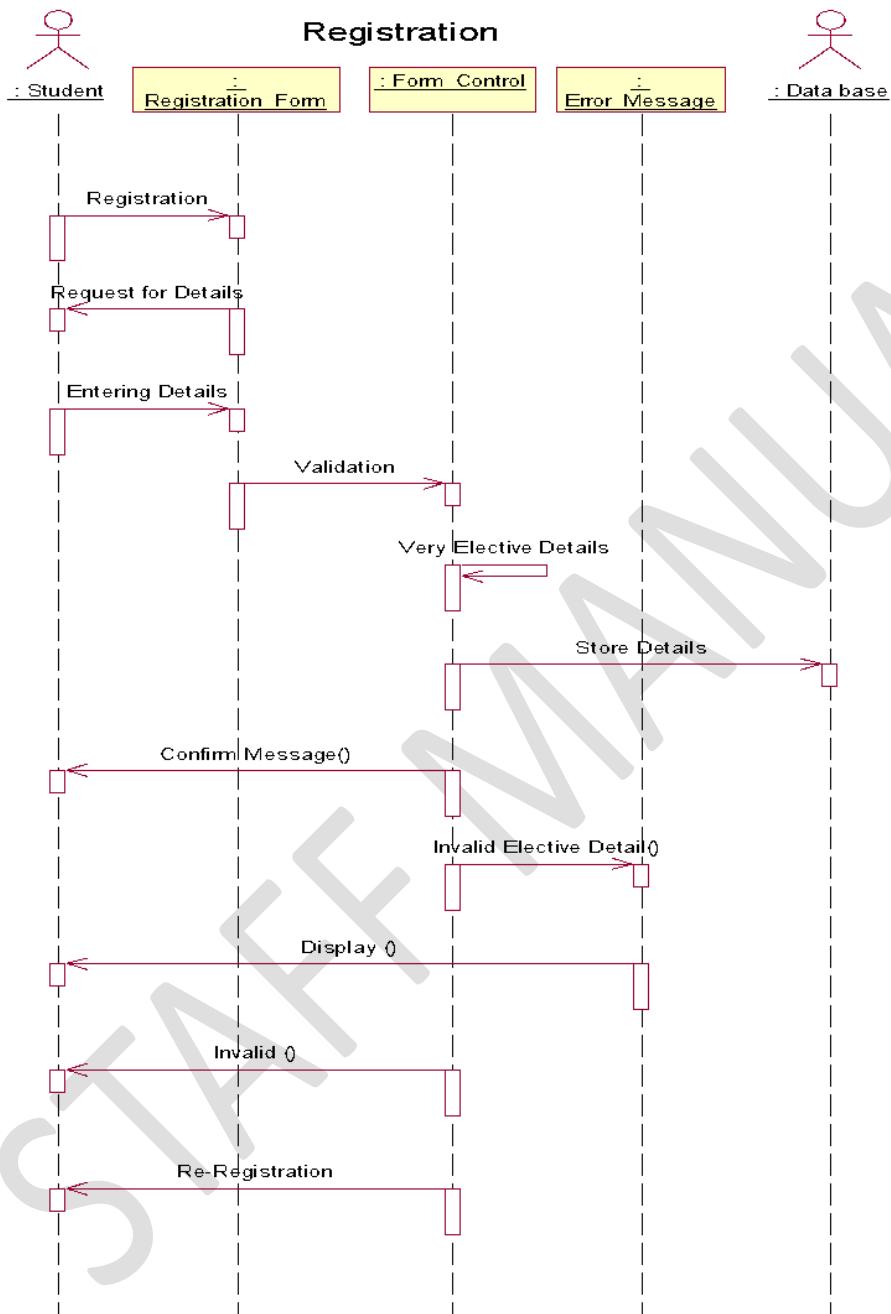
Activity Diagram:

STAFF MANUAL

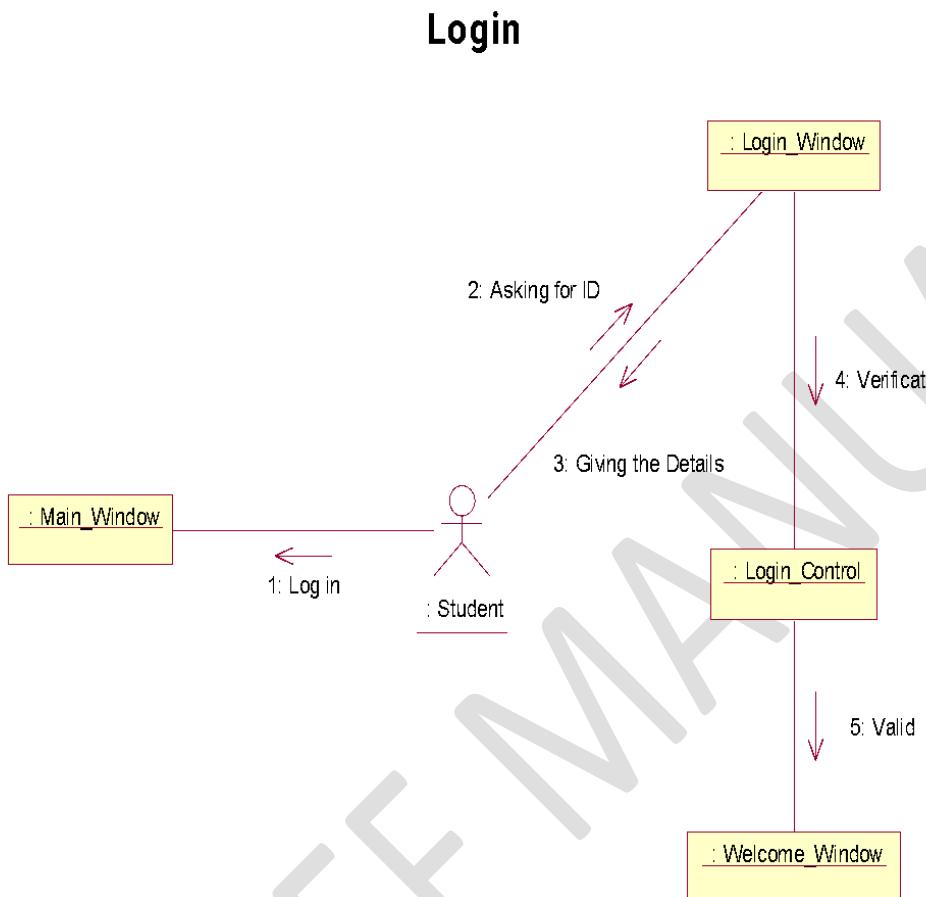
Sequence Diagrams:

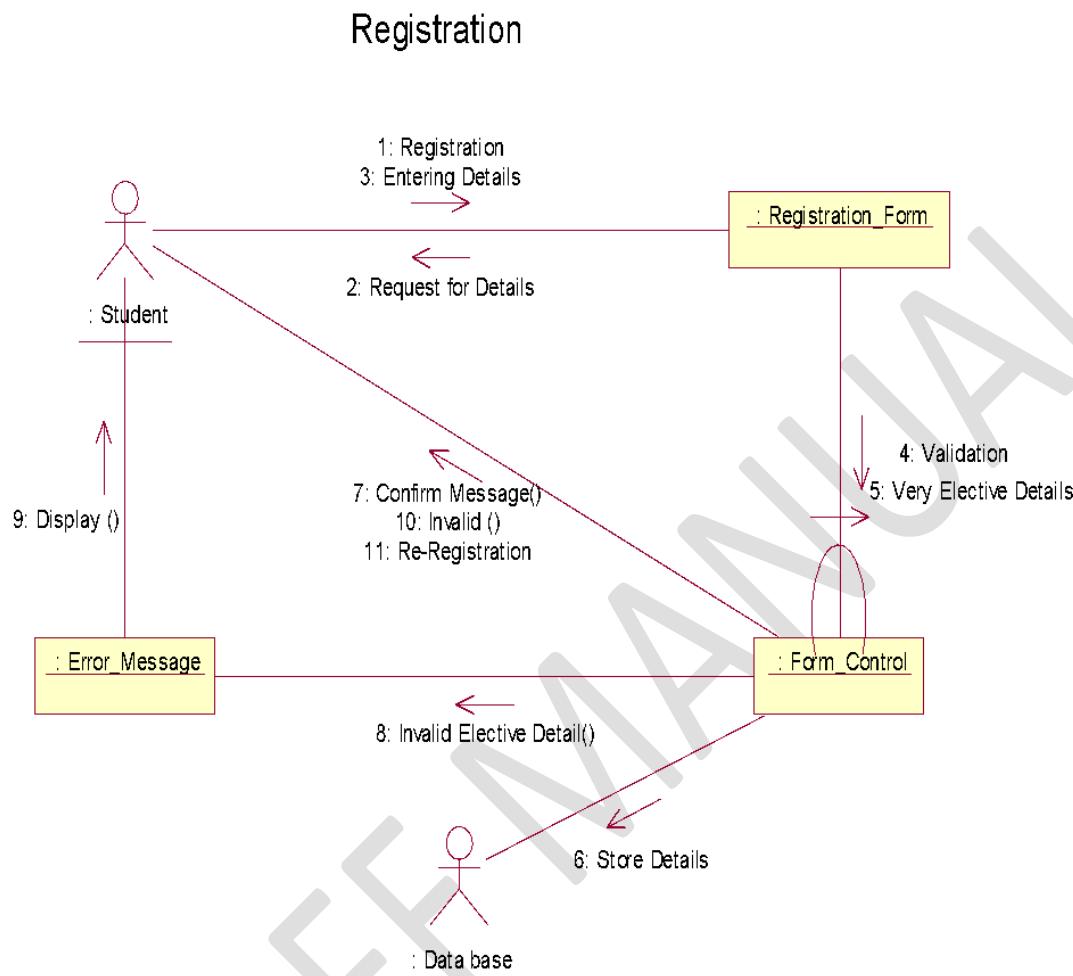




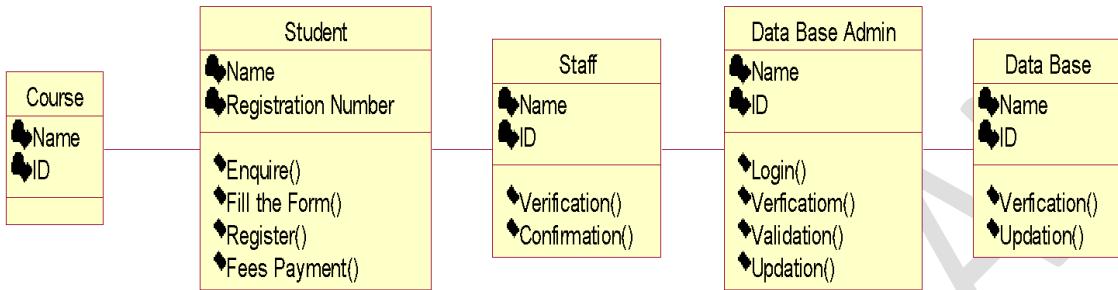


Collaborations Diagrams:



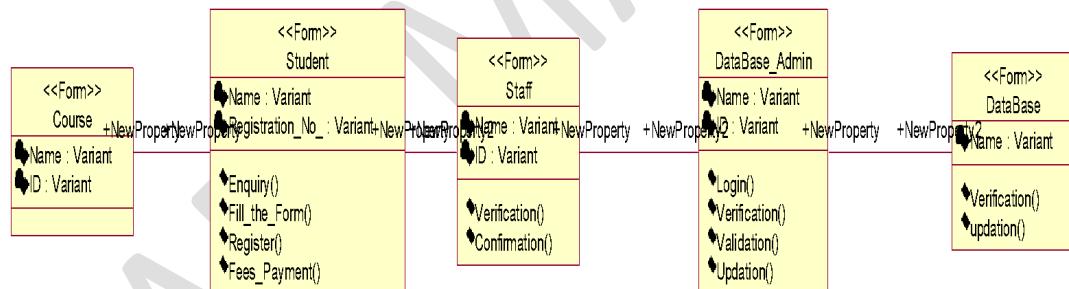


Class Diagram



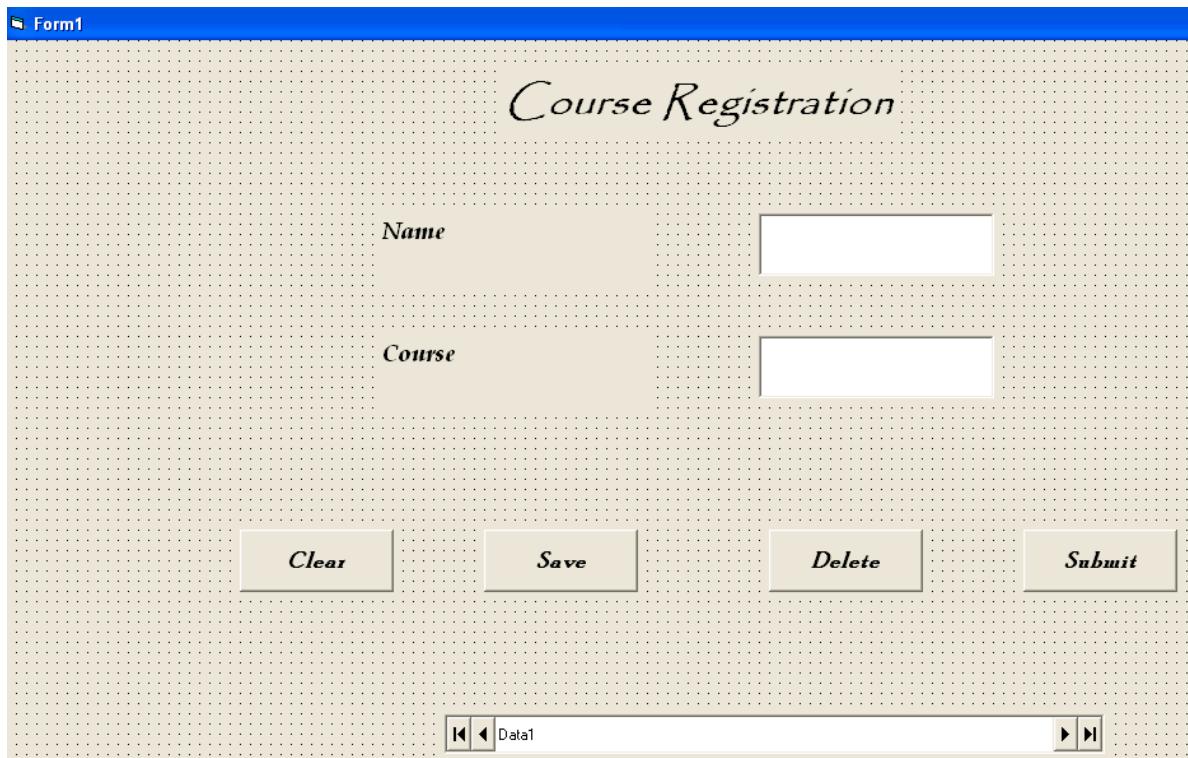
FORWARD ENGINEERING (Class Diagram)

Class Diagram



Form Design:

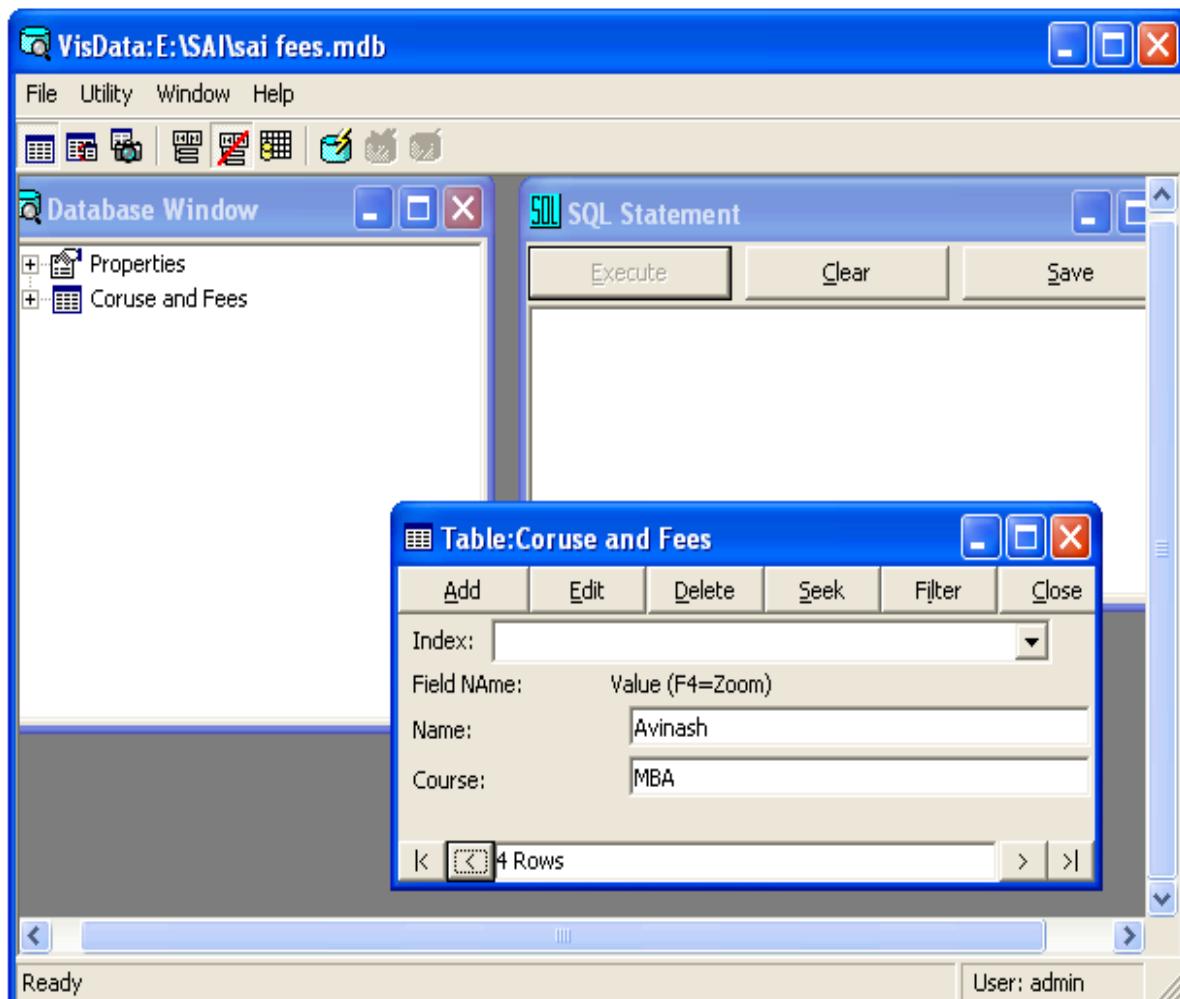
Form1:



Form2:

The screenshot shows a Windows application window titled "Form2". The main title bar has a blue header with the text "Form2". Below the title bar, the window has a light gray background with a dotted pattern. In the center, the text "Fees Payment" is displayed in a bold, italicized font. To the left of the text, there are three labels: "Name", "Course", and "Amount", each followed by a text input field. To the right of the "Amount" label is another text input field. At the bottom left is a button labeled "Accept", and at the bottom right is a button labeled "Exit".

Database Design:



Coding:

Form1:

```
Private Sub Clear_Click()
Data1.Recordset.AddNew
Text1.Text = ""
Text2.Text = ""
End Sub
Private Sub Delete_Click()
Data1.Recordset.Delete
MsgBox "Deleted"
End Sub
Private Sub Save_Click()
Data1.Recordset.Update
Data1.Recordset.MoveLast
MsgBox "Saved"
End Sub
Private Sub Submit_Click()
Form2.Show
Form2.Text1.Text = Form1.Text1.Text
Form2.Text2.Text = Form1.Text2.Text
End Sub
Form2:
Private Sub Accept_Click()
MsgBox "Fees Paid"
End Sub
Private Sub Exit_Click()
End
End Sub
```

Output:

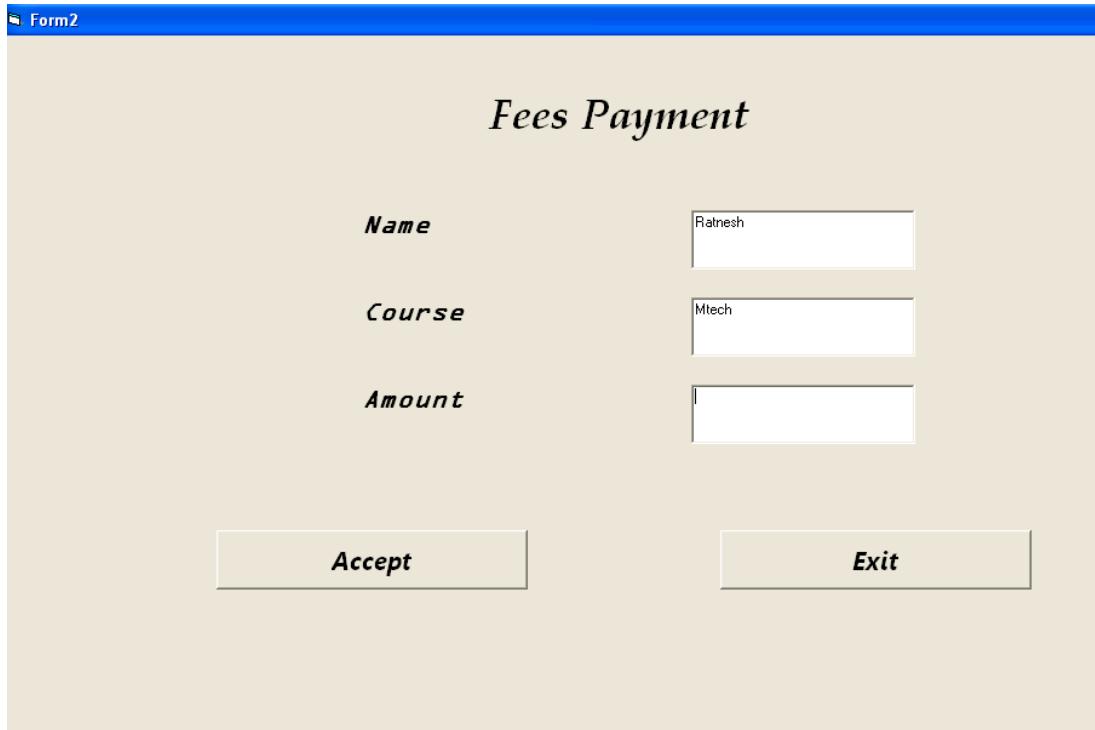
Form1

Course Registration

Name

Course

After Submission:



After Fees Payment:



Result:

Thus the course registration project is analyzed, designed using Rational Rose and implemented in VB.

EX.No:8

LIBRARY MANAGEMENT SYSTEM

AIM

To develop the Library Management System using rational rose tools, visual basic and MS access.

Introduction

To design a library management system using IBM Rational Rose. Implement this system using Visual Basic Language with Oracle/SQL.

Objective

Recently the importance of library management system within the service context of libraries has been unchallenged. Students check the list of books available and borrow the books if the book is a borrow book otherwise it is of waste for the student to come to the library to come to check for the books if the student doesn't get the book. Then the librarian checks the student id and allows the member to check out the book and the librarian then updates the member database and also the books database.

Scope

This system would be used by members who may be students or professors of that University to check the availability of the books and borrow the books, and by the librarian to update the databases. The purpose of this document is to analyze and elaborate on the high-level needs and features of the Online Library System. It focuses on the capabilities and facilities provided by a Library.

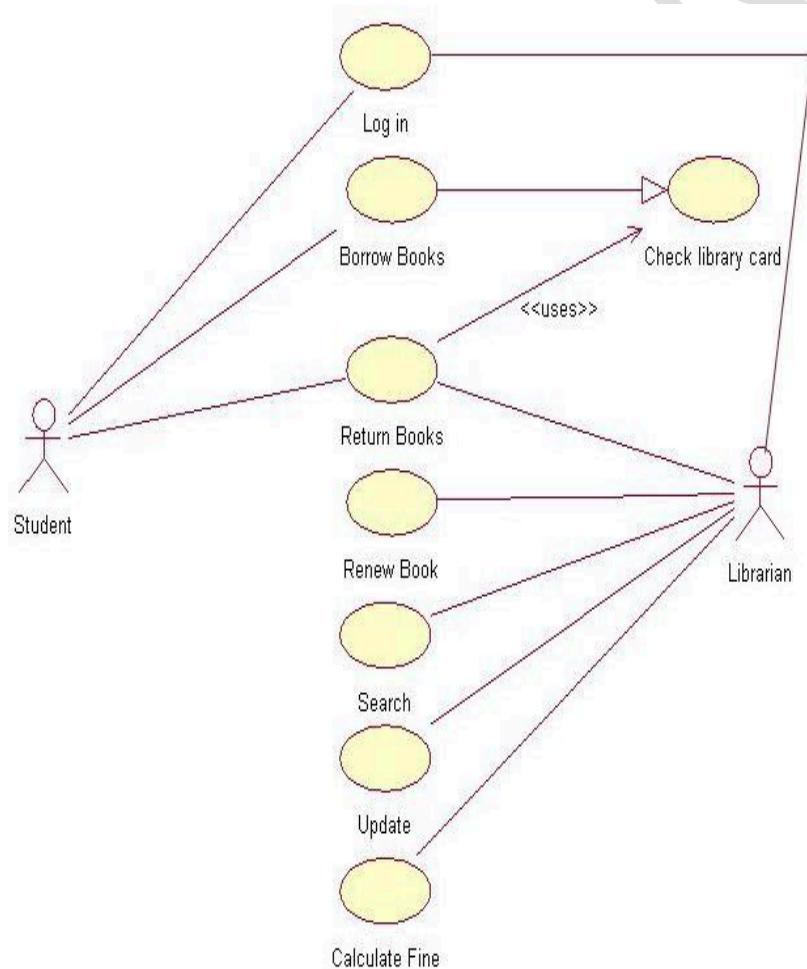
Problem Statement

The Library Management System is designed & developed for a receipt and issuance of books in the library along with the student's details. A librarian can add, delete and update book status and search from the database. A user can borrow, return books, reserve books and search for books. He can also renew his loan period. If a book is overdue, the user will be fined Rs.5 each day over the due. If a book is reported lost, the user will pay the full cost of the book. The books received in the library are entered in Books Entry form and the new student is entered in the student entry form. When the student wants to get the desired book the same is issued on the availability basis to the student. The issuance and due date for the returning of the book is also entered into the Book Issue form under third menu Book Issue.

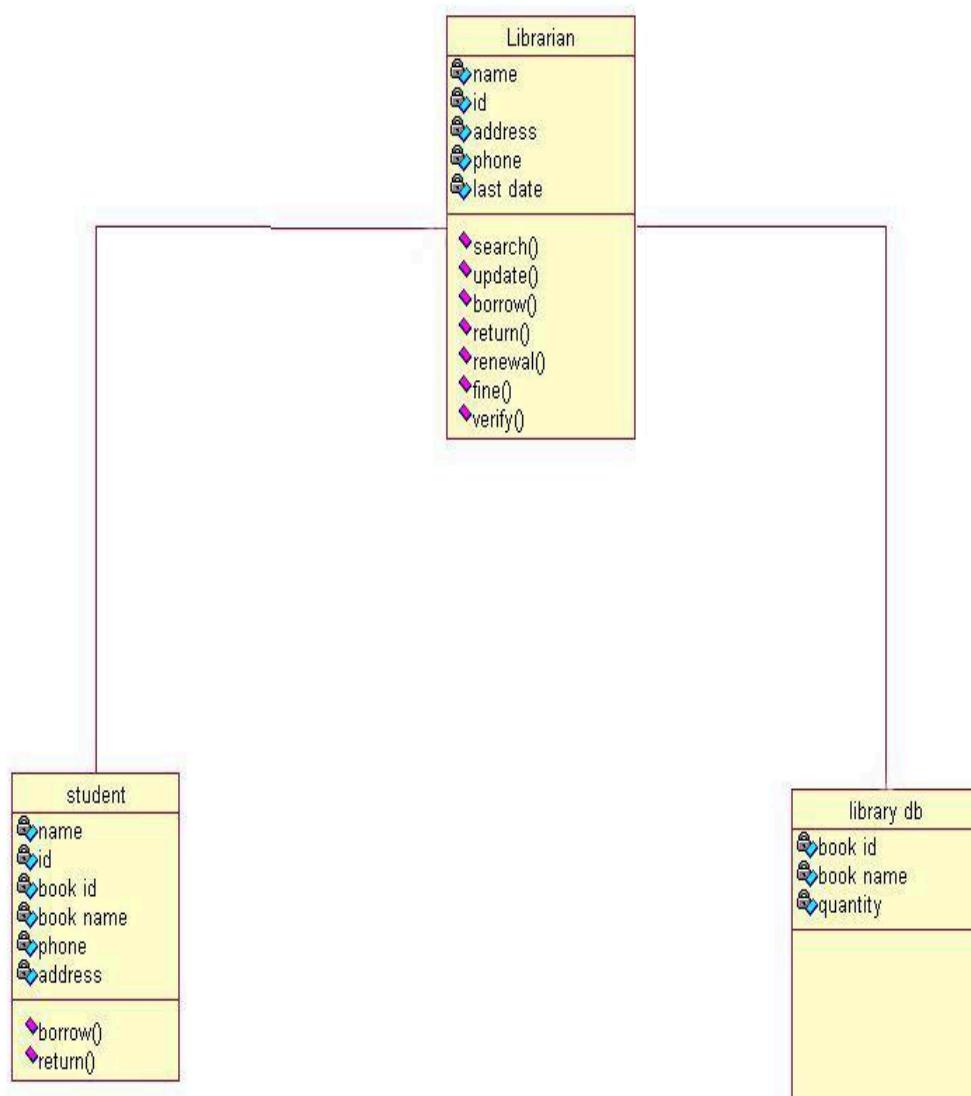
The student has to pay the fine if any on the basis of no. of days delayed deposit of the book in the library.

UML DIAGRAM

USECASE DIAGRAM:-

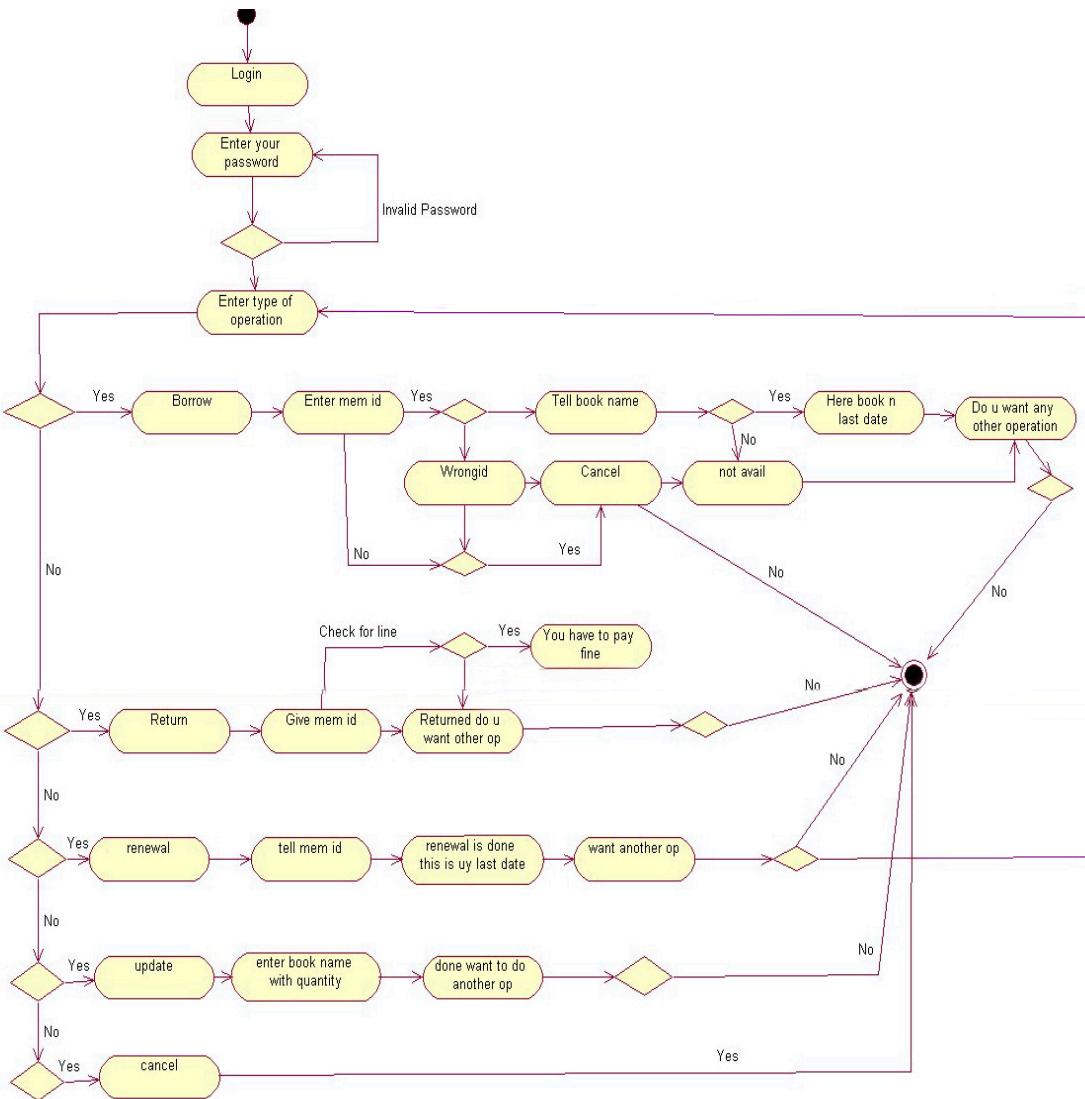


CLASS DIAGRAM



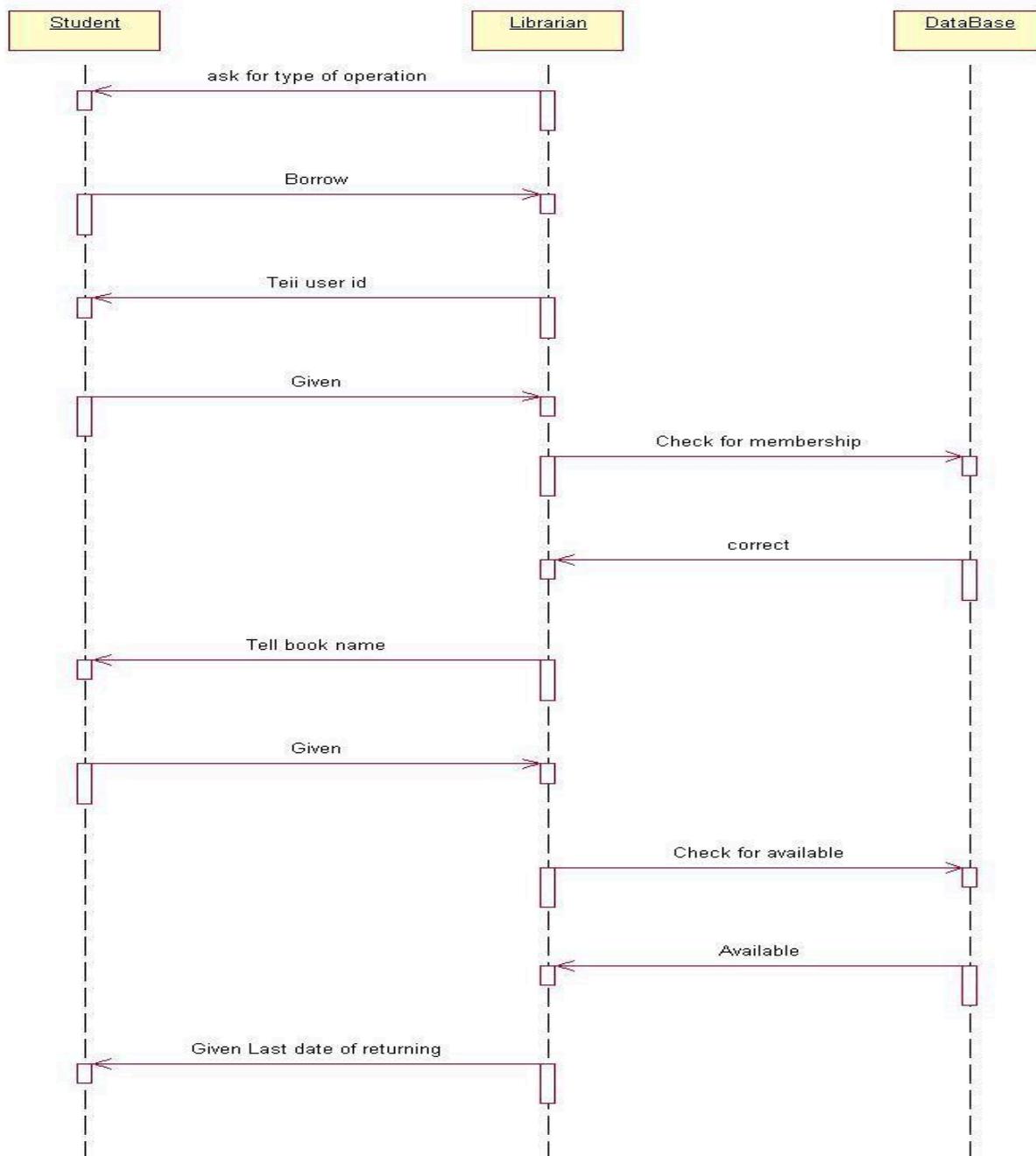
ACTIVITY DIAGRAM

STAFF MANUAL



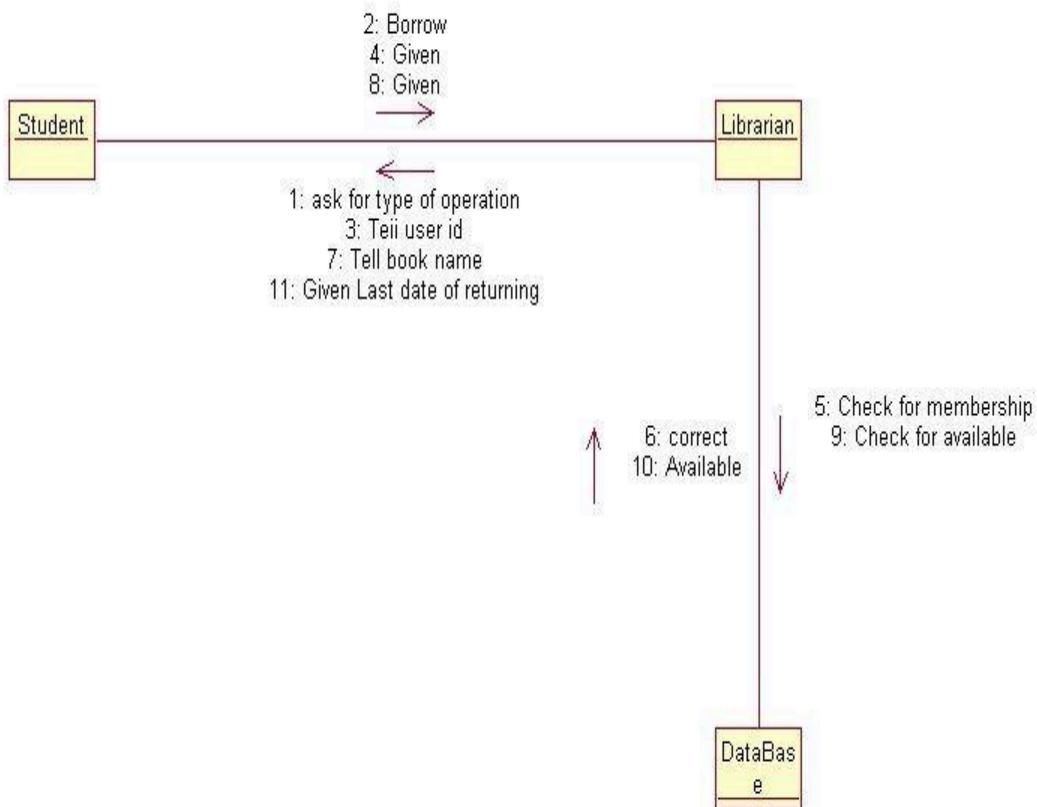
SEQUENCE DIAGRAM

CASE: ISSUE OF BOOKS



COLLABORATION DIAGRAM

CASE: ISSUE OF BOOKS



FORM DESIGN

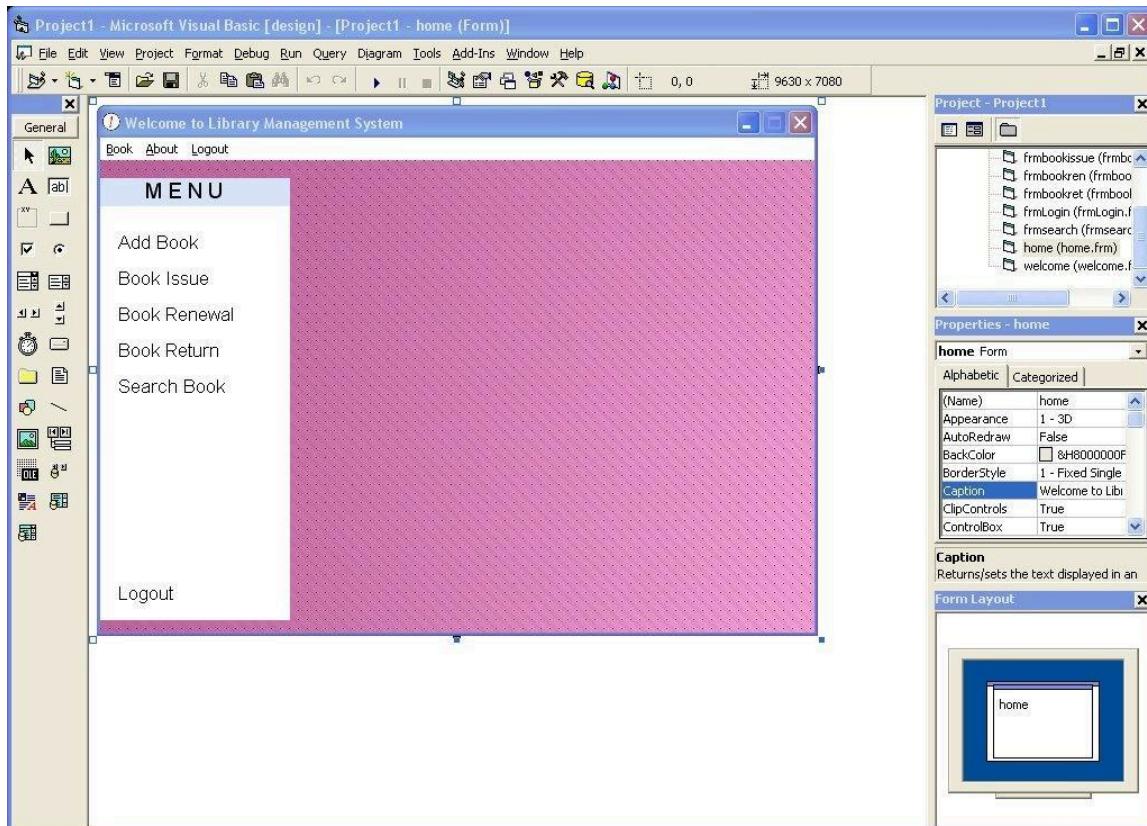
WELCOME PAGE:



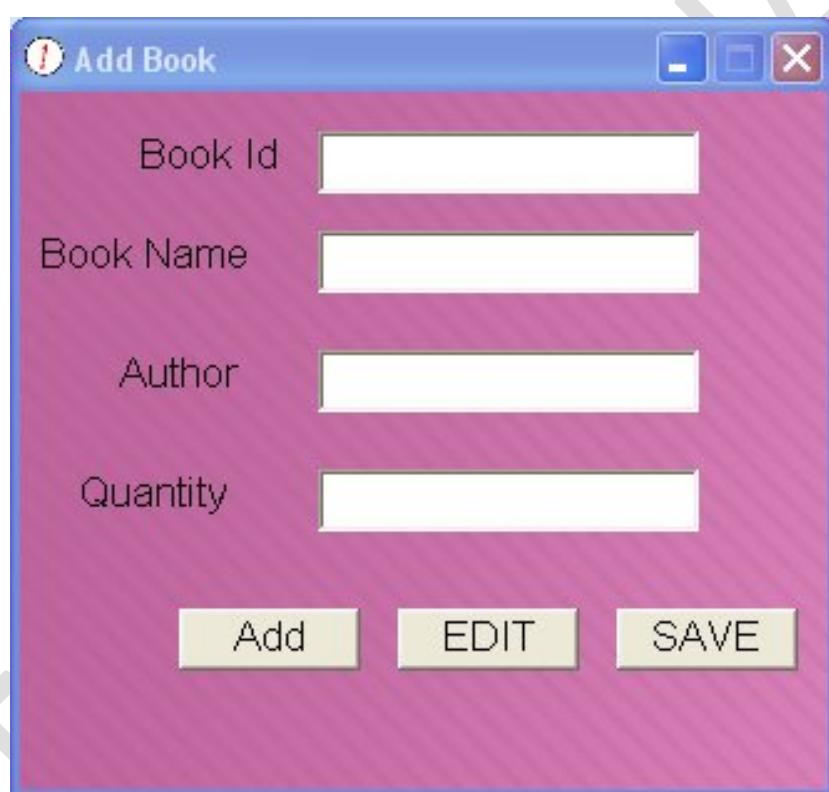
LOGIN PAGE:

The login page has a pink gradient background. It features two input fields: one for "User" and one for "Password", both with black outlines. Below these fields are two buttons: "Login" on the left and "Cancel" on the right, also with black outlines. A small green progress bar is visible at the bottom center.

MAIN MENU:



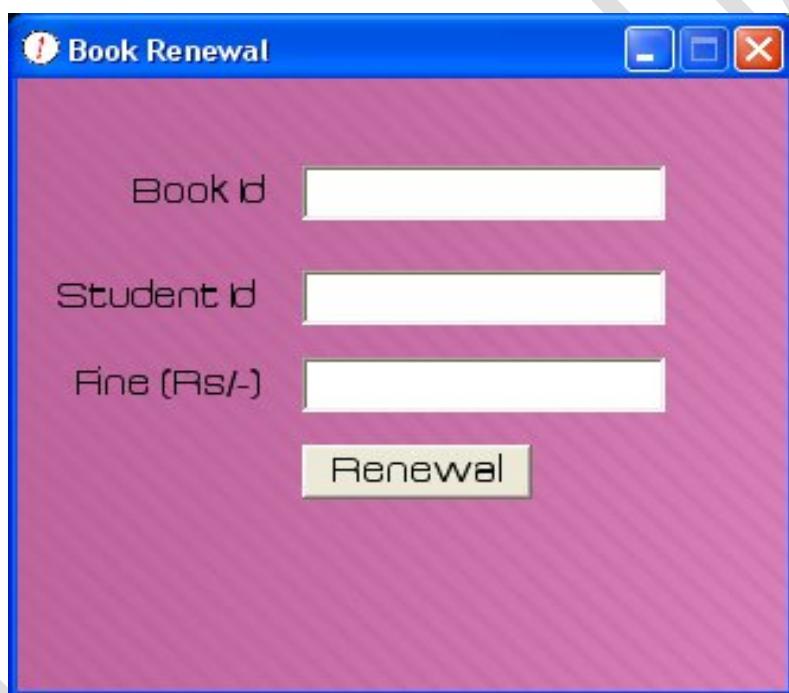
ADD BOOK



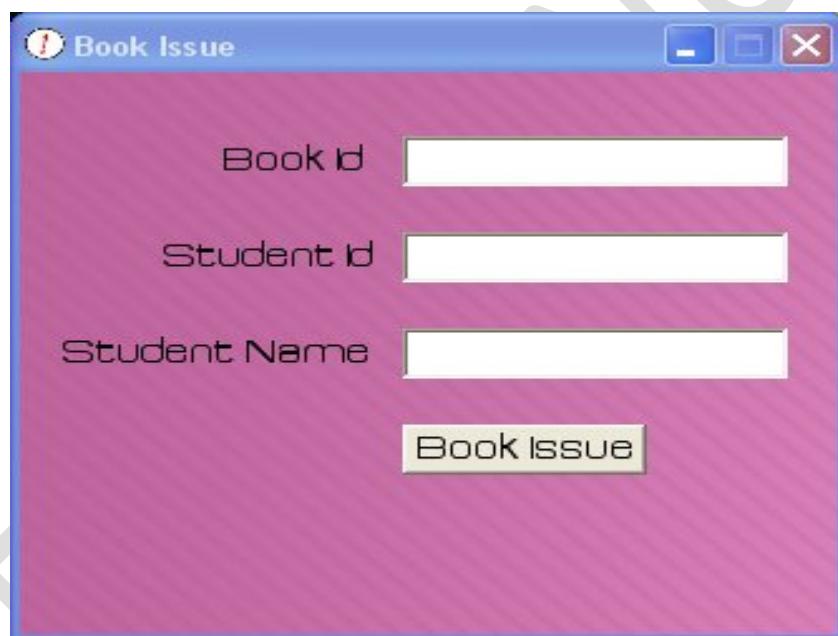
SEARCH PAGE:



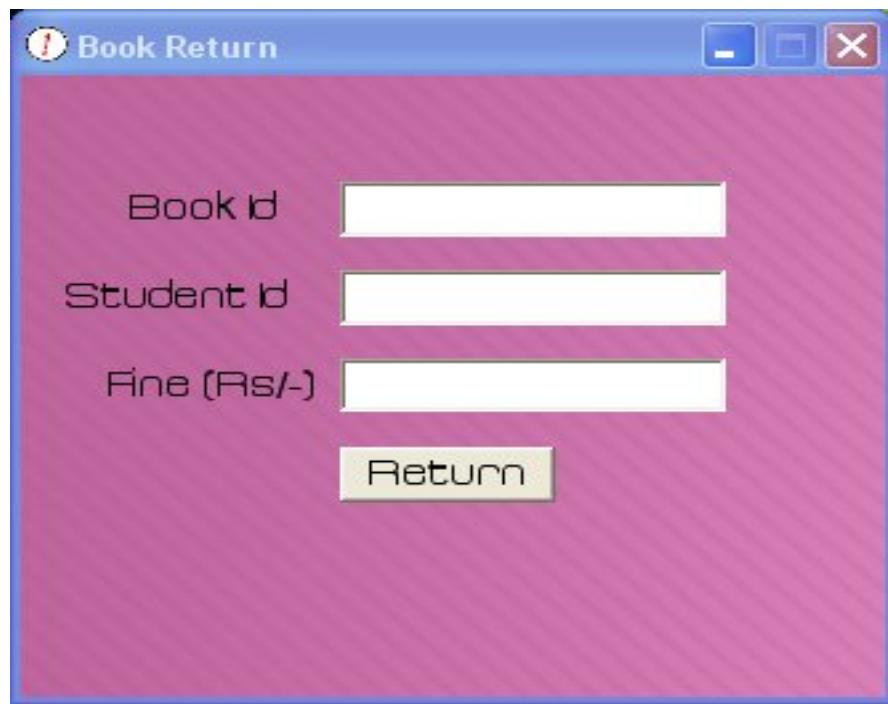
BOOK RENEWAL:



BOOK ISSUE:



BOOK RETURN:



DATABASE DESIGN:

BOOKS TABLE



Oracle SQL*Plus

File Edit Search Options Help

SQL> desc books;

Name	Null?	Type
ID	NOT NULL	NUMBER
NAME		VARCHAR2(100)
AUTHOR		VARCHAR2(50)
QUANTITY		NUMBER

SQL>

ISSUE:

The screenshot shows the Oracle SQL*Plus interface. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The main area displays the result of the SQL command "SQL> desc issue;". The output is a table with three columns: "Name", "Null?", and "Type". The data rows are as follows:

Name	Null?	Type
B_ID		NUMBER
S_ID		VARCHAR2(50)
S_NAME		VARCHAR2(250)
I_DATE		DATE
RET		VARCHAR2(1)

Below the table, there is a single line of text "SQL> |".

STA

CODING:

LOGIN FORM CODE:-

Option Explicit

Public LoginSucceeded As Boolean

Private Sub Cancel_Click()

'set the global var to false

'to denote a failed login

LoginSucceeded = False

End

End Sub

Private Sub Login_Click()

'check for correct password

If txtUserName = "admin" And txtPassword = "indian" Then

'place code to here to pass the

'success to the calling sub

'setting a global var is the easiest

LoginSucceeded = True

Unload Me

home.Show

Else

'Showing the Invalid Password message to user

```
Label1.Caption = "Invalid Password, try again!"  
  
txtUserName.Text = ""  
  
txtPassword.Text = ""  
  
End If  
  
End Sub
```

HOME FORM CODE:-

```
Dim cn As ADODB.Connection  
  
Dim rs As ADODB.Recordset  
  
Private Sub about_Click(Index As Integer)  
  
frmabout.Show  
  
End Sub  
  
Private Sub Add_Book_Click(Index As Integer)  
  
frmaddbook.Show  
  
End Sub  
  
Private Sub Book_Issue_Click(Index As Integer)  
  
frmbookissue.Show  
  
End Sub  
  
Private Sub Book_Renewal_Click(Index As Integer)  
  
frmbookren.Show  
  
End Sub  
  
Private Sub Book_Return_Click(Index As Integer)  
  
frmbookret.Show  
  
End Sub
```

```
Private Sub Label2_Click()
    frmaddbook.Show
End Sub

Private Sub Label3_Click()
    frmbookissue.Show
End Sub

Private Sub Label4_Click()
    frmbookren.Show
End Sub

Private Sub Label5_Click()
    frmbookret.Show
End Sub

Private Sub Label6_Click()
    LoginSucceeded = False
    Unload Me
    frmLogin.Show
End Sub

Private Sub Label7_Click()
    frmsearch.Show
End Sub

Private Sub logout_Click(Index As Integer)
    LoginSucceeded = False
    Unload Me
```

```
frmLogin.Show  
End Sub  
  
Private Sub Serch_Book_Click(Index As Integer)  
frmsearch.Show  
End Sub  
  
Private Sub Form_Load()  
Dim ss As String  
Set cn = New ADODB.Connection  
cn.Open "dsn=lib;uid=system;pwd=manager"  
Set rs = New ADODB.Recordset  
rs.Open ("select sum(quantity) as t_books from books"), cn, adOpenDynamic,  
adLockOptimistic  
Label8.Caption = "Available Books:: " & rs!t_books  
End Sub  
  
ADD & EDIT BOOK FORM CODE:-  
Dim cn As ADODB.Connection  
Dim rs As ADODB.Recordset  
Private Sub Add_Click()  
Set rs = cn.Execute("INSERT INTO books(id,name,author,quantity) VALUES(" & Text0.Text & ","  
& Text1.Text & "','" & Text2.Text & "','" & Text3.Text & ")")  
MsgBox "THE BOOK IS SAVED"  
'SET TEXT BOX EMPTY  
Text0.Text = ""
```

```
Text1.Text = ""  
  
Text2.Text = ""  
  
Text3.Text = ""  
  
End Sub  
  
Private Sub Edit_Click()  
  
If Text0.Text = "" Then  
  
    MsgBox "ENTER THE BOOK ID", vbInformation, " Data Required"  
  
    Exit Sub  
  
End If  
  
Set rs = cn.Execute("SELECT *from books where id=" & Text0.Text)  
  
If rs.EOF <> True Then  
  
    Text1.Text = rs(1)  
  
    Text2.Text = rs(2)  
  
    Text3.Text = rs(3)  
  
Else  
  
    MsgBox "Record Doesn't Exist"  
  
End If  
  
End Sub  
  
Private Sub Save_Click()  
  
If Text0.Text = "" Then  
  
    MsgBox "ENTER THE BOOK ID", vbInformation, " Data Required"  
  
    Exit Sub  
  
Else
```

```
cn.Execute ("update books set name="" & Text1.Text & "",author="" & Text2.Text &
 "",quantity="" & Text3.Text & "" where id=" & Val(Text0.Text))
```

```
MsgBox "BOOK SAVED"
```

```
'SET TEXT BOX EMPTY
```

```
Text0.Text = ""
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim ss As String
```

```
Set cn = New ADODB.Connection
```

```
cn.Open "dsn=lib;uid=system;pwd=manager"
```

```
End Sub
```

BOOK ISSUE FORM CODE:-

```
Dim cn As ADODB.Connection
```

```
Dim rs As ADODB.Recordset
```

```
Private Sub Issue_Click()
```

```
If Text0.Text = "" Then
```

```
MsgBox "Insert Book Id", vbInformation, " Data Required"
```

```
Exit Sub
```

```
End If
```

```
If rs.State Then
    rs.Close
End If

rs.Open "select * from books where id=" & Text0.Text & " ", cn, adOpenDynamic,
adLockOptimistic

If rs.EOF = False Then
    If rs(3) = 0 Then
        MsgBox "Books are not available"
        Exit Sub
    End If

    cn.Execute ("update books set quantity=quantity-1 where id=" & Text0.Text)

    'cn.Execute ("insert into issue set b_id=" & Text0.Text & ",s_id=" & Text1.Text & ",s_name=" &
    Text2.Text & ",i_date=""" & Format(Now, ("dd-mmm-yyyy")) & """)

If rs.State Then
    rs.Close
End If

rs.Open "select * from issue", cn, adOpenDynamic, adLockOptimistic

rs.AddNew

rs(0) = Text0.Text
rs(1) = Text1.Text
rs(2) = Text2.Text
rs(3) = Format(Now, ("dd-mmm-yyyy"))

rs.Update

MsgBox "THE BOOK IS ISSUED"
```

'SET TEXT BOX EMPTY

Text0.Text = ""

Text1.Text = ""

Text2.Text = ""

Else

MsgBox "Invalid Book Id"

End If

End Sub

Private Sub Form_Load()

Dim ss As String

Set cn = New ADODB.Connection

cn.Open "dsn=lib;uid=system;pwd=manager"

Set rs = New ADODB.Recordset

End Sub

BOOK RENEWAL FORM CODE:-

Dim cn As ADODB.Connection

Dim rs As ADODB.Recordset

Private Sub Renewal_Click()

Dim extra_days As Integer

Dim fine As Integer

If Text0.Text = "" Then

MsgBox "Insert Book Id", vbInformation, " Data Required"

```
Exit Sub

End If

rs.Open "select * from issue where b_id=" & Text0.Text & " and s_id =" & Text1.Text & " and
ret='n'", cn, adOpenDynamic, adLockOptimistic

If rs.EOF = False Then

    extra_days = DateDiff("d", rs(3), Format(Now, ("dd-mmm-yyyy")))

    If extra_days > 10 Then

        fine = (extra_days - 10) * 2

        'Also, use an editable text box so the correct amount a member
        'is payed is recorded. Sometimes the member may pay money not
        'exactly as required (payable $15 from $15.25 total fines)

        Text2.Text = fine

    Else

        Text2.Text = "0"

    End If

    If rs.State Then

        rs.Close

    End If

    'UPDATING ISSUE D DATE

    cn.Execute ("update issue set i_date=" & Format(DateAdd("d", 10, Now), ("dd-mmm-yyyy"))
    & " where b_id=" & Text0.Text & " and s_id =" & Text1.Text & " ")

    MsgBox " Renewed Book"

Else

    MsgBox "No Issues With Given Data"
```

```
End If

End Sub

Private Sub Form_Load()

Dim ss As String

Set cn = New ADODB.Connection

cn.Open "dsn=lib;uid=system;pwd=manager"

Set rs = New ADODB.Recordset

End Sub
```

BOOK RETURN FORM CODE:-

```
Dim cn As ADODB.Connection

Dim rs As ADODB.Recordset

Private Sub Return_Click()

Dim extra_days As Integer

Dim fine As Integer

If rs.State Then

rs.Close

End If

If Text0.Text = "" Then

MsgBox "Insert Book Id", vbInformation, " Data Required"

Exit Sub

End If
```

```
rs.Open "select * from issue where b_id=" & Text0.Text & " and s_id ="" & Text1.Text & "" and  
ret='n'", cn, adOpenDynamic, adLockOptimistic

If rs.EOF = False Then

    extra_days = DateDiff("d", rs(3), Format(Now, ("dd-mmm-yyyy")))

    If extra_days > 10 Then

        fine = (extra_days - 10) * 2

        'Also, use an editable text box so the correct amount a member  

        'is payed is recorded. Sometimes the member may pay money not  

        'exactly as required (payable $15 from $15.25 total fines)

        Text2.Text = fine

    Else

        Text2.Text = "0"

    End If

    If rs.State Then

        rs.Close

    End If

    cn.Execute ("update issue set ret='y' where b_id=" & Text0.Text & " and s_id ="" & Text1.Text &  
""")

    If rs.State Then

        rs.Close

    End If

    cn.Execute ("update books set quantity=quantity+1 where id=" & Text0.Text)

    MsgBox " The book " & Text0.Text & " has been returned from " & Text1.Text

Else
```

MsgBox "No Issues With This Data"

End If

End Sub

Private Sub Form_Load()

Dim ss As String

Set cn = New ADODB.Connection

cn.Open "dsn=lib;uid=system;pwd=manager"

Set rs = New ADODB.Recordset

End Sub

BOOK SEARCH FORM CODE:-

Dim cn As ADODB.Connection

Dim rs As ADODB.Recordset

Private Sub Search_Click()

Dim max As Integer

Dim i As Integer

List1.Clear

If Text0.Text = "" Then

MsgBox "Please enter the book name", vbInformation, " Data Required"

Exit Sub

End If

```
If rs.State Then
    rs.Close
End If

rs.Open ("select *from books where name like "" & Text0.Text & "%""), cn, adOpenDynamic,
adLockOptimistic

If rs.EOF = False Then
    rs.MoveFirst
    Do While Not rs.EOF
        List1.AddItem rs(1) & " - " & rs(0)
        rs.MoveNext
    Loop
Else
    MsgBox "No Matches"
End If

End Sub

Private Sub Form_Load()
    Dim ss As String
    Set cn = New ADODB.Connection
    cn.Open "dsn=lib;uid=system;pwd=manager"
    Set rs = New ADODB.Recordset
End Sub
```

OUTPUT

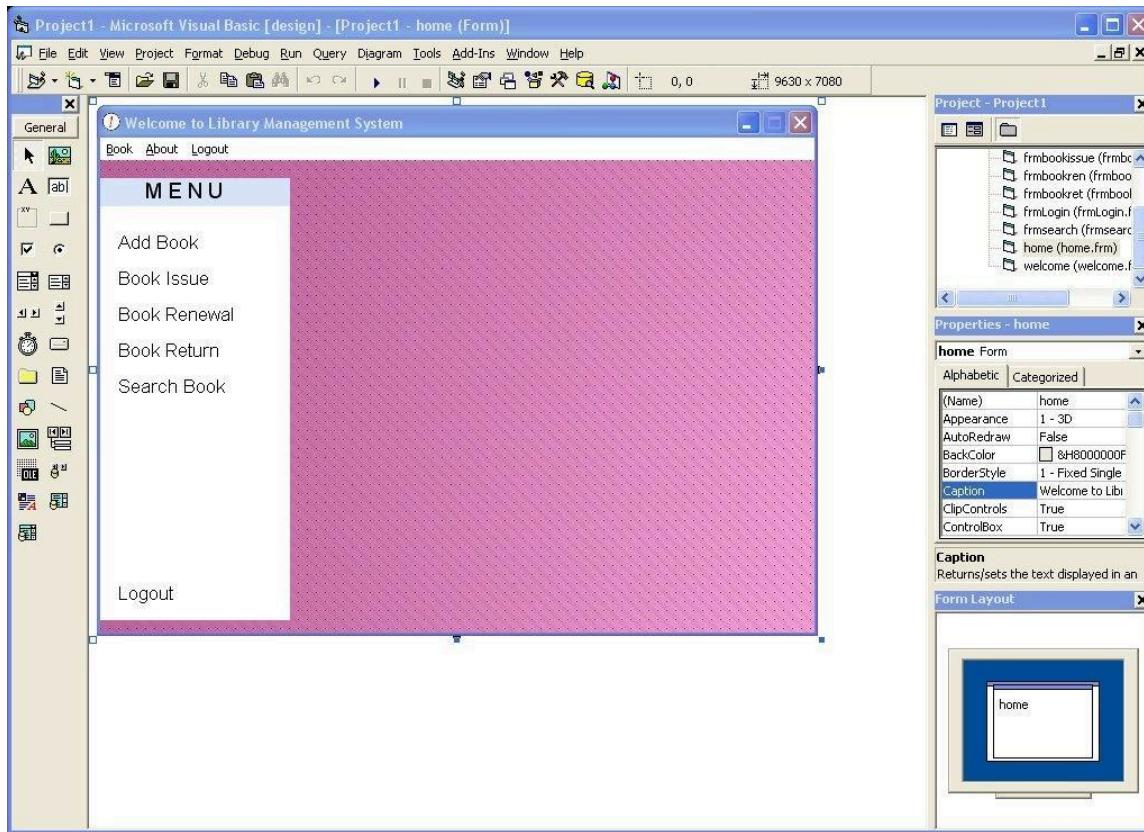
WELCOME PAGE



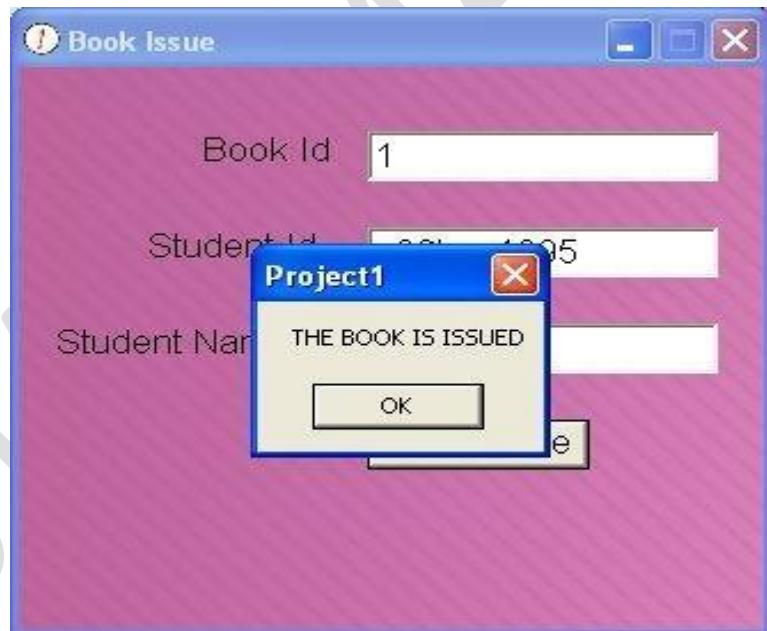
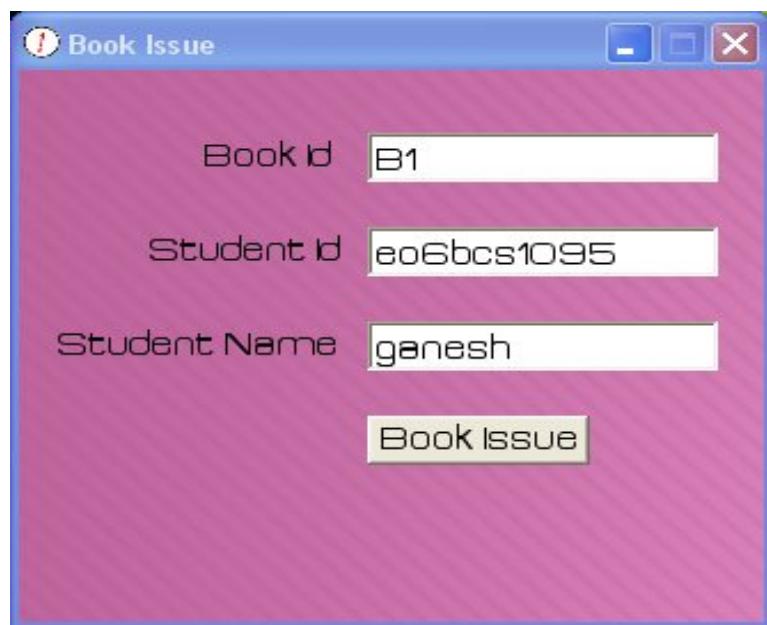
LOGIN PAGE:

The login page has a pink background. It features two input fields: one for "User" containing "admin" and another for "Password" containing five asterisks. Below the password field is a small green progress bar. At the bottom are two buttons: "Login" on the left and "Cancel" on the right.

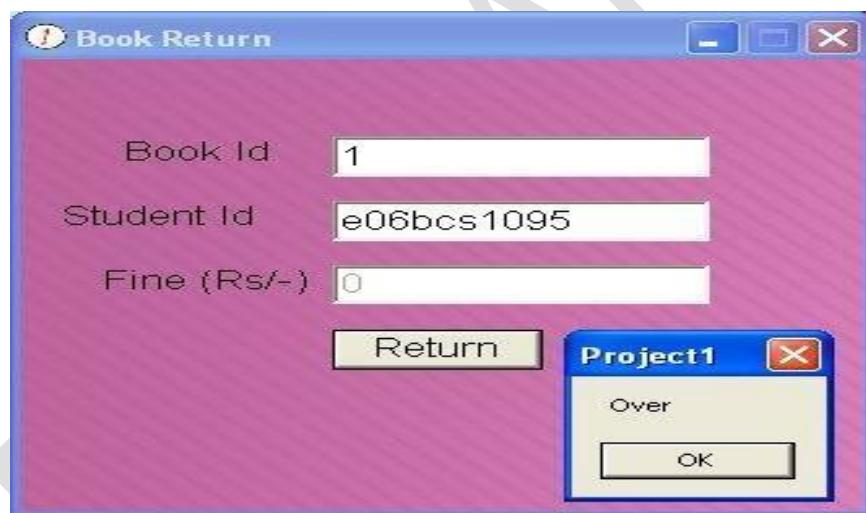
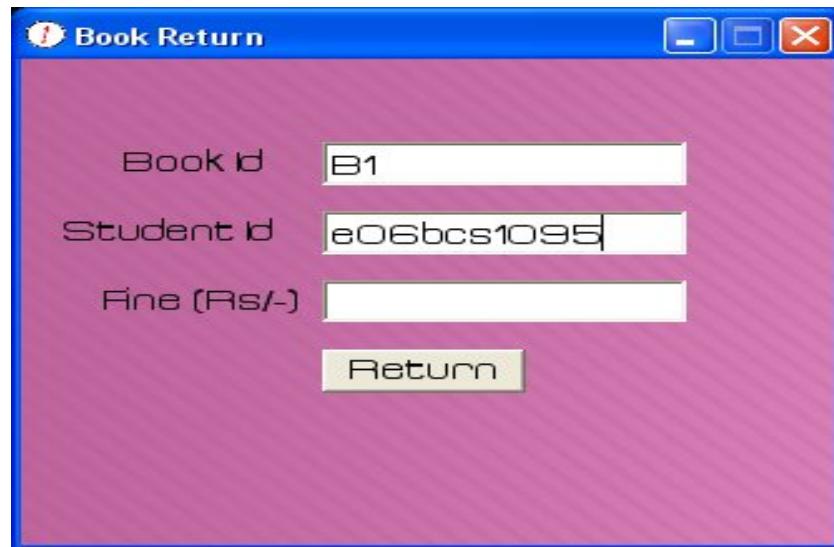
MAIN MENU



BOOK ISSUE



BOOK RETURN



RESULT:

Thus the Library Management System is designed and implemented successfully and the output is verified.

EX.NO :9

PASSPORT AUTOMATION SYSTEM

AIM

To develop the Passport Automation System using rational rose tools, visual basic and MS access.

Objective:

To develop the passport automation system software using UML language. It is the interface between applicant and authority responsible for issue the passport. It aims at improving efficiency and reducing complexities.

Scope:

The system provides the online interface to the user where we can fill their form and personal detail with necessary proof. The authority concerned with the issue of passport can use this system to reduce its workload and process it speedy manner. It provides communication platform between administrator and applicant. To transfer the data between passport authority and local police verifying the applicant's information.

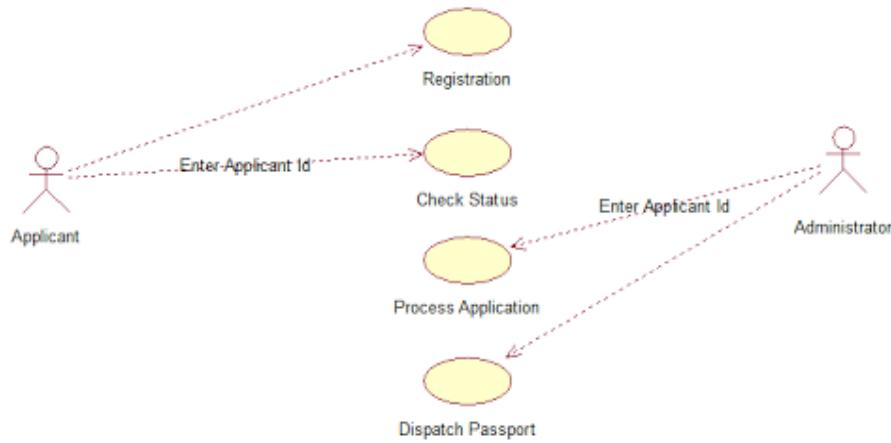
Problem Statement

My project title is passport automation system. In this project we can develop the 5 modules such as login, fill the form with necessary proof, verifying the applicant's information, validity checking and issue the passport for that particular applicant.

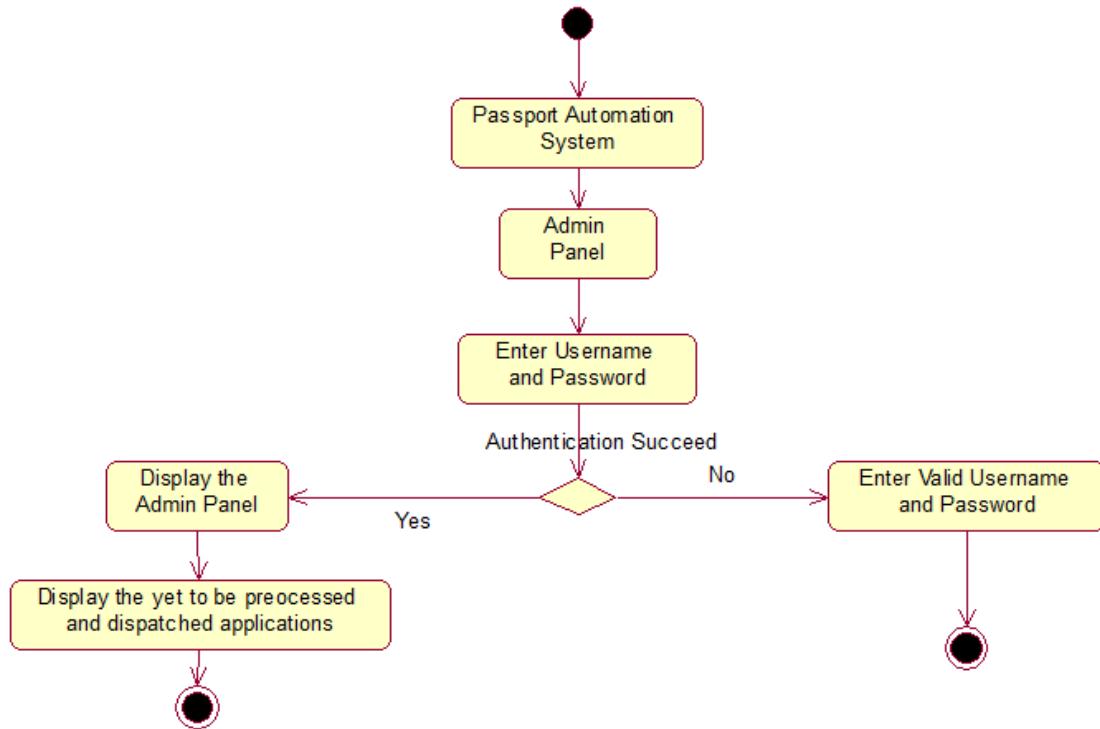
In this login module, we can perform that enters into the login website for the different actors, and then fill for the can be done by the primary and verification, checking and issuing the passport can be done for the supporting actor.

UML Diagram

Use Case Diagram



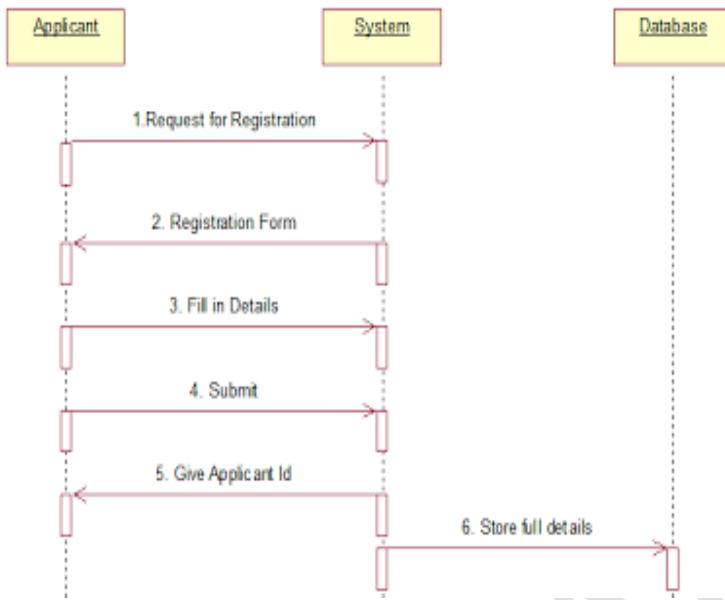
Activity Diagram



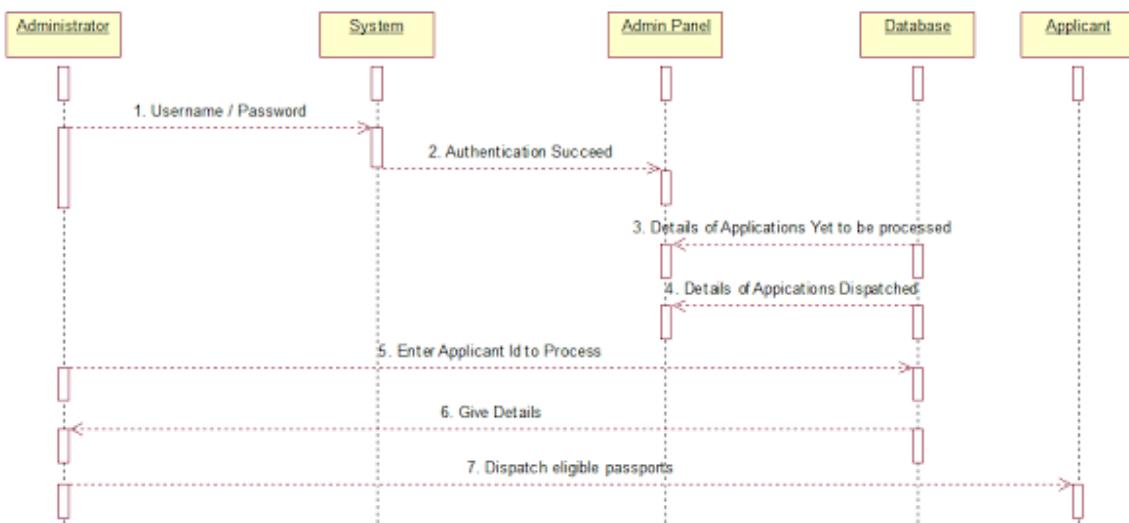
Sequence Diagram

New Registration

Sequence diagram for New Registration

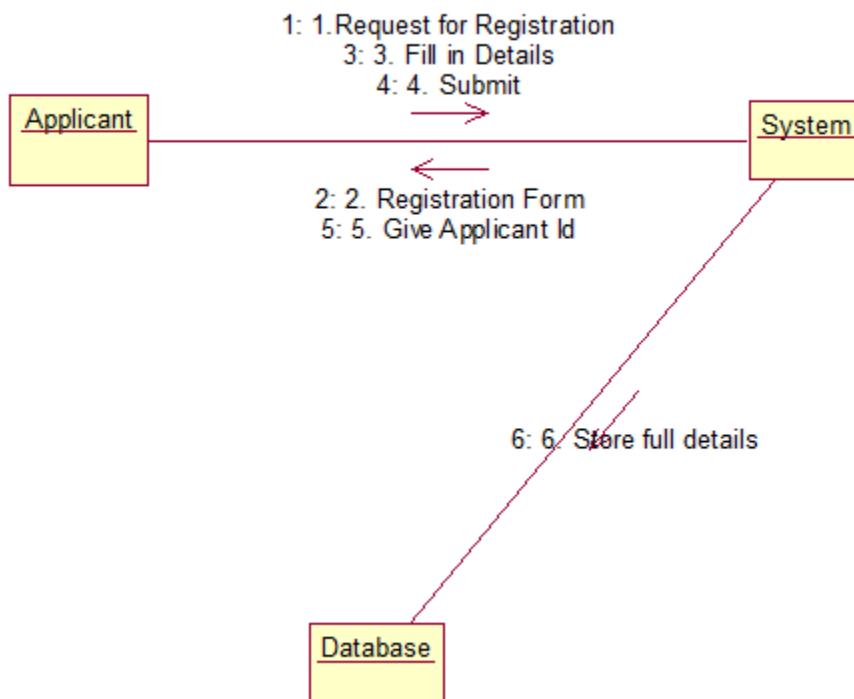


Admin Panel

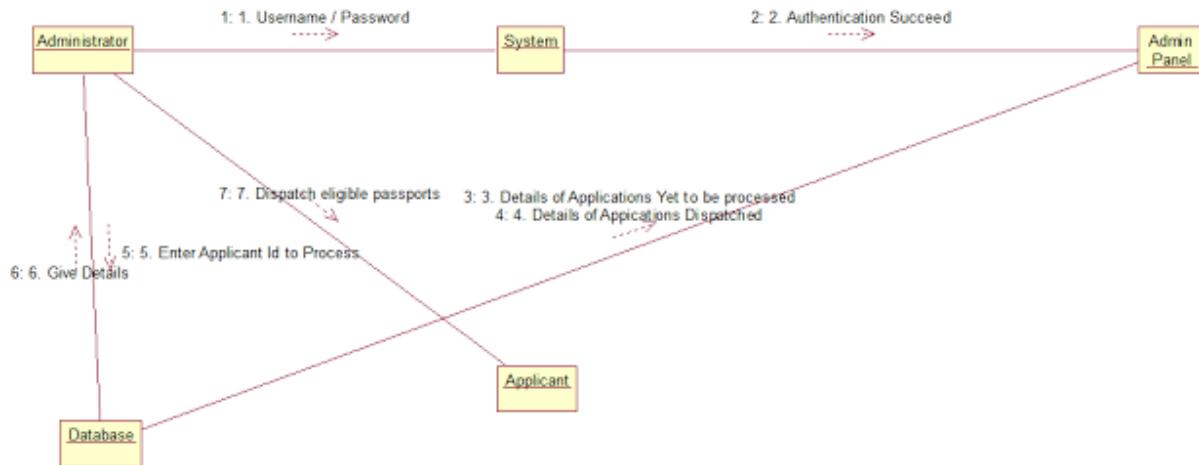


Collaboration Diagram

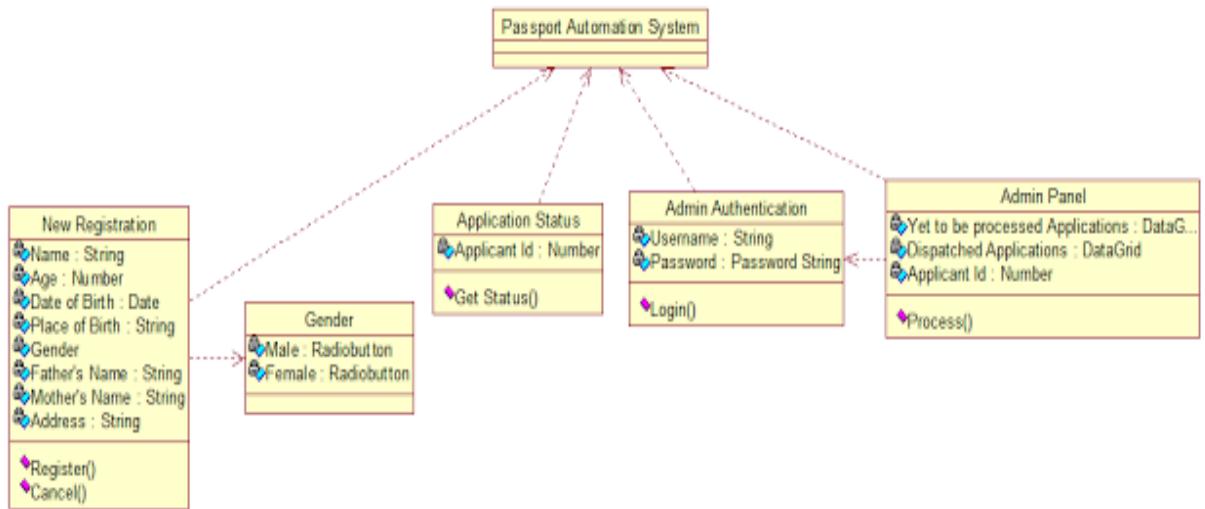
New Registration



Admin Panel

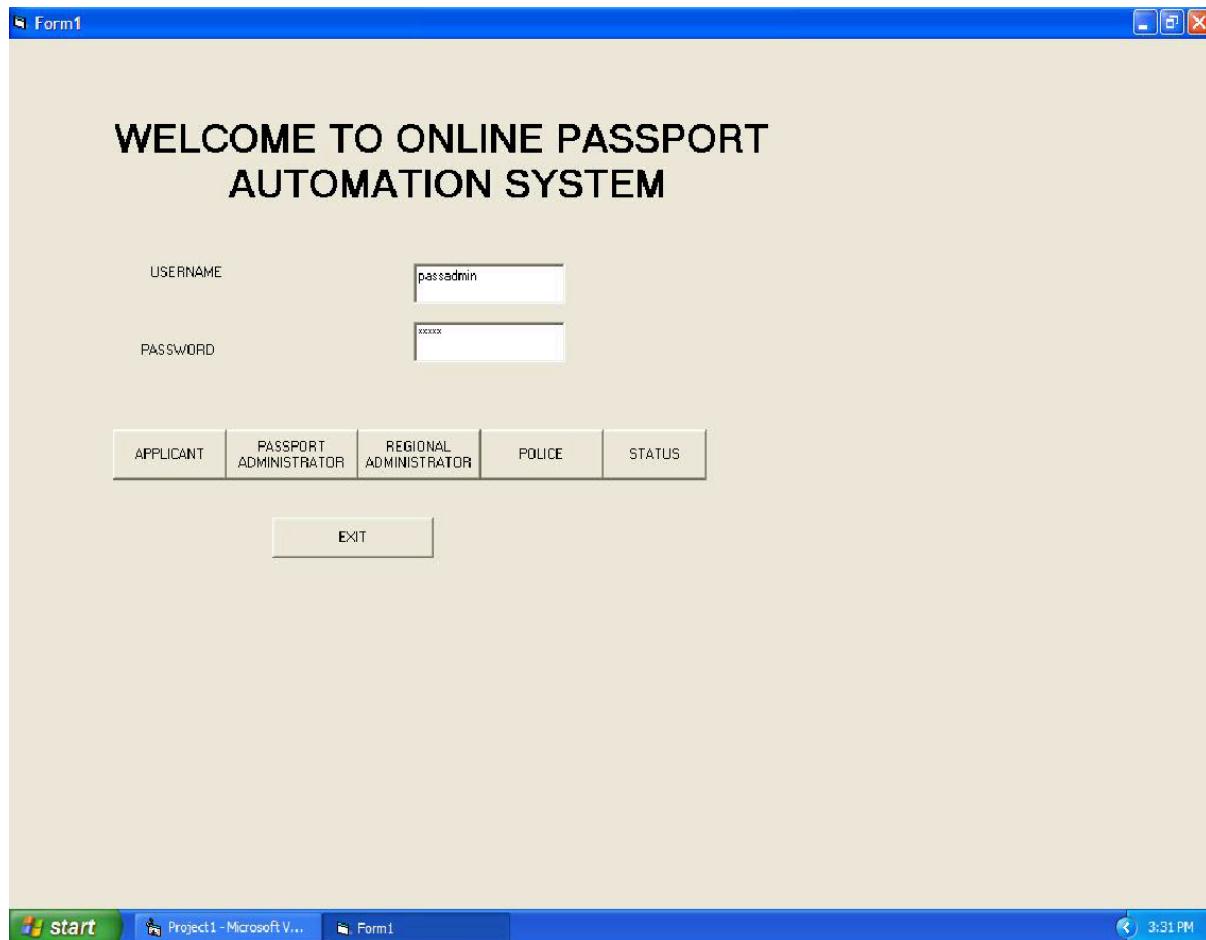


Class Diagram



FORM DESIGN

FORM 1



FORM2

Form2

GIVE YOUR DETAILS

NAME	palani	
FATHERNAME	pathasarathi	
DATE OF BIRTH	27/2/1990	SUBMIT
PERMANENT ADDRESS	neelangarai	CANCEL
TEMPORARY ADDRESS	neelangarai	
PHONE NO	9445310441	
EMAILID	palani@gmail.c	◀ ▶ Data1
PAN	1000	

SOURCE CODE:

FORM1

```
Private Sub Command1_Click()
Dim app As Applicant
Set app = New Applicant
app.Login
End Sub

Private Sub Command2_Click()
Dim pass As PassportAdministrator
Set pass = New PassportAdministrator
pass.Login
End Sub

Private Sub Command3_Click()
Dim reg As RegionalAdministrator
Set reg = New RegionalAdministrator
reg.Login
End Sub

Private Sub Command4_Click()
Dim pol As Police
Set pol = New Police
pol.Login
End Sub

Private Sub Command5_Click()
If Form1.Text1.Text = "" And Form1.Text2.Text = "" Then
    MsgBox "LOGIN SUCCESSFUL"
    Form6.Show
Else
    MsgBox "INVALID USERNAME AND PASSWORD"
    Unload Me
End If
End Sub

Private Sub Command6_Click()
End
End Sub
```

FORM2:

```
Private Sub Command1_Click()
Dim subdetails As Applicant
Set subdetails = New Applicant
subdetails.SubmitDetails
End Sub
Private Sub Command3_Click()
Data1.Recordset.Edit
End Sub
Private Sub Command4_Click()
Data1.Recordset.update
End Sub
Private Sub Form_Load()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
End Sub
```

FORM3:

```
Private Sub a_Click()
Data2.Recordset.AddNew
End Sub
Private Sub Command1_Click()
Dim search As PassportAdministrator
Set search = New PassportAdministrator
search.update
End Sub
Private Sub Command2_Click()
If Data1.Recordset.BOF Then
MsgBox "NO DATA FOUND"
Else
Data1.Recordset.MovePrevious
End If
End Sub
Private Sub Command3_Click()
```

```
If Data1.Recordset.EOF Then
    MsgBox "NO DATA FOUND"
Else
    Data1.Recordset.MoveNext
End If
End Sub
Private Sub Command4_Click()
    Form1.Show
    Unload Me
End Sub
Private Sub Command5_Click()
    Data1.Recordset.MoveFirst
End Sub
Private Sub Command6_Click()
    Data1.Recordset.MoveLast
End Sub
Private Sub Command7_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(9) = "successful"
    Data1.Recordset.update
End Sub
Private Sub Command8_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Fields(9) = "unsuccessful"
    Data1.Recordset.update
End Sub
Private Sub ve_Click()
    Dim verify As PassportAdministrator
    Set verify = New PassportAdministrator
    verify.update
End Sub
FORM4:
Private Sub Command1_Click()
    Dim search As RegionalAdministrator
    Set search = New RegionalAdministrator
    search.verify
End Sub
Private Sub Command2_Click()
```

```
Data1.Recordset.Edit
Data1.Recordset.Fields(10) = "successful"
Data1.Recordset.update
End Sub
Private Sub Command3_Click()
Data1.Recordset.Edit
Data1.Recordset.Fields(10) = "unsuccessful"
Data1.Recordset.update
End Sub
Private Sub Command4_Click()
Form1.Show
Unload Me
End Sub
Private Sub Command5_Click()
Dim update As RegionalAdministrator
Set update = New RegionalAdministrator
update.update
End Sub
Private Sub Command6_Click()
Data1.Recordset.MoveLast
End Sub
Private Sub Command7_Click()
Data1.Recordset.MoveFirst
End Sub
Private Sub Command8_Click()
If Data1.Recordset.BOF Then
MsgBox "NO DATA FOUND"
Else
Data1.Recordset.MovePrevious
End If
End Sub
Private Sub Command9_Click()
If Data1.Recordset.EOF Then
MsgBox "NO DATA FOUND"
Else
Data1.Recordset.MoveNext
End If
End Sub
```

FORM5:

```
Private Sub Command1_Click()
Dim search As Police
Set search = New Police
search.verify
End Sub

Private Sub Command2_Click()
Data2.Recordset.Edit
Data2.Recordset.Fields(11) = "successful"
Data2.Recordset.update
End Sub

Private Sub Command3_Click()
Data2.Recordset.Edit
Data2.Recordset.Fields(11) = "unsuccessful"
Data2.Recordset.update
End Sub

Private Sub Command4_Click()
Form1.Show
Unload Me
End Sub

Private Sub Command6_Click()
Data2.Recordset.MoveLast
End Sub

Private Sub Command7_Click()
Data2.Recordset.MoveFirst
End Sub

Private Sub Command8_Click()
If Data2.Recordset.BOF Then
MsgBox "NO DATA FOUND"
Else
Data2.Recordset.MovePrevious
End If
End Sub

Private Sub Command9_Click()
If Data2.Recordset.EOF Then
MsgBox "NO DATA FOUND"
Else
Data2.Recordset.MoveNext
End If
End Sub
```

End If

End Sub

FORM6:

```
Private Sub Command1_Click()
Dim checkstate As Applicant
Set checkstate = New Applicant
checkstate.CheckStatus
End Sub
Private Sub Command2_Click()
Form1.Show
Unload Me
End Sub
```

UML CODINGS:

APPLICANT:

```
Option Explicit
'##ModelId=4D7521E80271
Private Name As Variant
'##ModelId=4D7521EE02CE
Private FatherName As Variant
'##ModelId=4D7521F3009C
Private DateOfBirth As Variant
'##ModelId=4D75220B01A5
Private PermanentAddress As Variant
'##ModelId=4D752220033C
Private TemporaryAddress As Variant
'##ModelId=4D75224101E4
Private EmailID As Variant
'##ModelId=4D75224701E4
Private PhoneNumber As Variant
'##ModelId=4D75224E031C
Private PanNo As Variant
'##ModelId=4D7522590242
Private ApplicationNo As Variant
'##ModelId=4D75225D038A
Private UserName As Variant
'##ModelId=4D75226300CB
Private Password As Variant
'##ModelId=4D7523300271
```

```
Public NewProperty As Database
'##ModelId=4D7522690109
Public Sub Login()
If Form1.Text1.Text = "" And Form1.Text2.Text = "" Then
MsgBox "LOGIN SUCCESSFUL"
Form2.Show
Else
MsgBox "INVALID USERNAME AND PASSWORD"
Unload Me
End If
End Sub
'##ModelId=4D752271032C
Public Sub SubmitDetails()
Dim ap As Integer
Form1.Data1.Recordset.MoveLast
ap = Data1.Recordset.Fields(0)
Data1.Recordset.AddNew
ap = ap + 1
Form1.Data1.Recordset.Fields(0) = ap
Form1.Data1.Recordset.Fields(1) = Text1.Text
Form1.Data1.Recordset.Fields(2) = Text2.Text
Form1.Data1.Recordset.Fields(3) = Text3.Text
Form1.Data1.Recordset.Fields(4) = Text4.Text
Form1.Data1.Recordset.Fields(5) = Text5.Text
Form1.Data1.Recordset.Fields(6) = Text6.Text
Form1.Data1.Recordset.Fields(7) = Text7.Text
Form1.Data1.Recordset.Fields(8) = Text8.Text
Form1.Data1.Recordset.Fields(9) = "Under Process"
Form1.Data1.Recordset.Fields(10) = "Under Process"
Form1.Data1.Recordset.Fields(11) = "Under Process"
Form1.Data1.Recordset.update
Form1.Show
Unload Me
End Sub
'##ModelId=4D7522760261
Public Sub CheckStatus()
Dim Currentdb As Database
Set Currentdb = OpenDatabase("D:\PASSPORTPROJECT1\PASSPORT.mdb")
```

Dim Data As Recordset

Set Data = Currentdb.OpenRecordset("applicant", dbOpenDynaset)

Data.FindFirst "([ApplicationNo])=" & Form6.Text1.Text

If Data.NoMatch Then

MsgBox "No such record"

Else

MsgBox "success"

Form6.Text1.Text = Data.Fields(0)

Form6.Label5 = Data.Fields(1)

Form6.Label7 = Data.Fields(9)

Form6.Label9 = Data.Fields(10)

Form6.Label11 = Data.Fields(11)

End If

Data.Close

End Sub

DATA BASE:

Option Explicit

'##ModelId=4D7522A30222

Private Name As Variant

'##ModelId=4D75233C005D

Public NewProperty As PassportAdministrator

'##ModelId=4D75233E006D

Public NewProperty2 As RegionalAdministrator

'##ModelId=4D75234202BF

Public NewProperty3 As Police

'##ModelId=4D7522A50186

Public Sub store()

End Sub

PASSPORT ADMINISTRATOR:

Option Explicit

'##ModelId=4D7522A90128

Private UserName As Variant

'##ModelId=4D7522F9035B

Private Password As Variant

'##ModelId=4D7522B20232

Public Sub Login()

If Form1.Text1.Text = "passadmin" And Form1.Text2.Text = "12345" Then

MsgBox "LOGIN SUCCESSFUL"

```
Form1.Text1.Text = ""  
Form1.Text2.Text = ""  
Form1.Text1.SetFocus  
Form3.Show  
Else  
MsgBox "INVALID USERNAME OR PASSWORD"  
Form1.Text1.Text = ""  
Form1.Text2.Text = ""  
Form1.Text1.SetFocus  
End If  
End Sub  
##ModelId=4D7522BA004E  
Public Sub verify()  
Set Currentdb = OpenDatabase("D:\PASSPORTPROJECT1\passport.mdb")  
Dim Data As Recordset  
Set Data = Currentdb.OpenRecordset("PassportAdministrator", dbOpenDynaset)  
If Form3.Text1.Text = "" Then  
MsgBox "select any data"  
Else  
Data.FindFirst "([PanNo])=" & Form3.Text1.Text  
If Data.NoMatch Then  
MsgBox "No such record"  
Else  
MsgBox "success"  
Form3.Text7.Text = Data.Fields(1)  
Form3.Text8.Text = Data.Fields(2)  
End If  
End If  
End Sub  
##ModelId=4D7522BF01D4  
Public Sub update()  
If Form3.Text2.Text = Form3.Text7.Text And Form3.Text3.Text =  
Form3.Text8.Text Then  
Form3.Data1.Recordset.Edit  
Form3.Data1.Recordset.Fields(9) = "successful"  
Form3.Data1.Recordset.update  
MsgBox "success"  
Else
```

```
MsgBox "no "
Form3.Text7.Text = ""
Form3.Text8.Text = ""
End If
End Sub
POLICE:
Option Explicit
##ModelId=4D7522E1001F
Private UserName As Variant
##ModelId=4D75232601D4
Private Password As Variant
##ModelId=4D7522E30251
Public Sub Login()
If Form1.Text1.Text = "poladmin" And Form1.Text2.Text = "12345" Then
MsgBox "LOGIN SUCCESSFUL"
Form1.Text1.Text = ""
Form1.Text2.Text = ""
Form1.Text1.SetFocus
Form5.Show
Else
MsgBox "INVALID USERNAME OR PASSWORD"
Form1.Text1.Text = ""
Form1.Text2.Text = ""
Form1.Text1.SetFocus
End If
End Sub
##ModelId=4D7522E8008C
Public Sub verify()
Dim Currentdb As Database
Set Currentdb = OpenDatabase("D:\PASSPORTPROJECT1\passport.mdb")
Dim Data As Recordset
Set Data = Currentdb.OpenRecordset("Police", dbOpenDynaset)
If Form5.Text1.Text = "" Then
MsgBox "select any data"
Else
Data.FindFirst "[PanNo] = " & Form5.Text1.Text
If Data.NoMatch Then
MsgBox "No such record"
```

```
Else
MsgBox "success"
If Data.Fields(5) = "Notallowed" Then
    MsgBox "not allowed"
Else
    MsgBox "allowed"
End If
End If
End If
Data.Close
End Sub
'##ModelId=4D7522EA02BF
```

```
Public Sub update()
End Sub
```

REGIONAL ADMINISTRATOR:

```
Option Explicit
'##ModelId=4D7522C80222
Private UserName As Variant
'##ModelId=4D75231A0109
Private Password As Variant
'##ModelId=4D7522CB02CE
Public Sub Login()
    If Form1.Text1.Text = "regadmin" And Form1.Text2.Text = "12345" Then
        MsgBox "LOGIN SUCCESSFUL"
        Form1.Text1.Text = ""
        Form1.Text2.Text = ""
        Form1.Text1.SetFocus
        Form4.Show
    Else
        MsgBox "INVALID USERNAME OR PASSWORD"
        Form1.Text1.Text = ""
        Form1.Text2.Text = ""
        Form1.Text1.SetFocus
    End If
End Sub
'##ModelId=4D7522CE01A5
Public Sub verify()
    Dim Currentdb As Database
```

```
Set Currentdb = OpenDatabase("D:\PASSPORTPROJECT1\passport.mdb")
Dim Data As Recordset
Set Data = Currentdb.OpenRecordset("RegionalAdministrator", dbOpenDynaset)
If Form4.Text1.Text = "" Then
    MsgBox "select any data"
Else
    Data.FindFirst "([PanNo])=" & Form4.Text1.Text
    If Data.NoMatch Then
        MsgBox "No such record"
    Else
        MsgBox "success"
        Form4.Text6.Text = Data.Fields(1)
        Form4.Text7.Text = Data.Fields(2)
        Form4.Text8.Text = Data.Fields(3)
        Form4.Text9.Text = Data.Fields(4)
    End If
End If
End Sub
'##ModelId=4D7522D002BF
Public Sub update()
If Form4.Text2.Text = Text6.Text And Form4.Text3.Text = Form4.Text7.Text
And Form4.Text4.Text = Form4.Text8.Text And Form4.Text5.Text =
Form4.Text9.Text Then
    MsgBox "Details match"
Else
    MsgBox "Details donot match"
End If
End Sub
```

OUT PUT

Form4

REGIONAL ADMINISTRATOR

APPLICATION NO	1	NAME	palani
PAN	1000	FATHER NAME	sarathi
NAME	palani	DOB	2/27/1990
FATHER NAME	sarathi	PERMANENT ADDRESS	neelangarai
DOB	2/27/1990	PERMANENT ADDRESS	neelangarai
PERMANENT ADDRESS	neelangarai		

SEARCH **VERIFY** **GRANT** **DONOT GRANT**

PREVIOUS **NEXT** **FIRST** **LAST**

CLOSE

start Project1 - Microsoft V... Form1 Form4 3:36 PM

Form5

POLICE

PAN	1000	APPLICATION NO	1																																										
NAME	palani																																												
<table border="1"><thead><tr><th>PanNo</th><th>Name</th><th>FatherName</th><th>DateOfBirth</th><th>PermanentA</th><th>FIR</th></tr></thead><tbody><tr><td>1000</td><td>palani</td><td>pathasarathi</td><td>2/27/1990</td><td>neelangarai</td><td>Allowed</td></tr><tr><td>1001</td><td>natraj</td><td>murali</td><td>11/2/1990</td><td>mambalam</td><td>Allowed</td></tr><tr><td>1002</td><td>pandi</td><td>raja</td><td>4/3/1990</td><td>madurai</td><td>Allowed</td></tr><tr><td>1003</td><td>prem</td><td>murugan</td><td>8/9/1991</td><td>ernore</td><td>Allowed</td></tr><tr><td>1004</td><td>karthi</td><td>rajendran</td><td>3/28/1990</td><td>medipakkam</td><td>Allowed</td></tr><tr><td>1005</td><td>anand</td><td>sathish</td><td>8/6/1990</td><td>kovai</td><td>Notallowed</td></tr></tbody></table>				PanNo	Name	FatherName	DateOfBirth	PermanentA	FIR	1000	palani	pathasarathi	2/27/1990	neelangarai	Allowed	1001	natraj	murali	11/2/1990	mambalam	Allowed	1002	pandi	raja	4/3/1990	madurai	Allowed	1003	prem	murugan	8/9/1991	ernore	Allowed	1004	karthi	rajendran	3/28/1990	medipakkam	Allowed	1005	anand	sathish	8/6/1990	kovai	Notallowed
PanNo	Name	FatherName	DateOfBirth	PermanentA	FIR																																								
1000	palani	pathasarathi	2/27/1990	neelangarai	Allowed																																								
1001	natraj	murali	11/2/1990	mambalam	Allowed																																								
1002	pandi	raja	4/3/1990	madurai	Allowed																																								
1003	prem	murugan	8/9/1991	ernore	Allowed																																								
1004	karthi	rajendran	3/28/1990	medipakkam	Allowed																																								
1005	anand	sathish	8/6/1990	kovai	Notallowed																																								

SEARCH GRANT DONT GRANT

PREVIOUS NEXT FIRST LAST

CLOSE

◀ ▶ □ ▢ ▣ ▤ ▥

start Project1 - Microsoft V... Form1 Form5 3:45 PM

Result

Thus the Passport Automation system is analyzed using Rational Rose and implemented using VB.