**Online Course Reservation System**

**Aim**

         To develop a problem statement, SRS document, UML Diagrams, Technical, Domain & Service layers for Online Course Reservation System.

**Problem Statement**

The Online Course Reservation System helps a student to reserve a course from a college of his or her choice online. If a student is already registered, they can log in. Else, they have to fill up the registration form provided by the administrator. After registering, the student can log in to the system. Then the student selects the college and courses. Then based on the qualifications of the student the system determines if the student is eligible for the course he has opted for. If the course is available the student can proceed to the payment step, where the course fee can be paid online. After paying the fee, a receipt stating the details of the student and course they selected is provided by the Administrator. Then the student successfully logs out the system.

**Result**

Thus the problem statement for Course Reservation system is written.

**GANTT CHART**

**AIM:**

To develop Gantt chart for Online Course Reservation System.

**GANTT CHART:**



**Result**

Thus the Gantt chart for Online Course Reservation System was successfully developed.

|  |  |
| --- | --- |
| Online Course Reservation System | Version: 1.0 |
| Modern Software Requirements Specification | Date: 4.1.12 |

Online Course Reservation System

Modern Software Requirements Specification

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 29.12.2011 | 1.0 | It contains the first major version of SRS. It contains the initial analysis of the system. | Yogeshwaran .G  Shahul Hameed .S  Siva Sankar .A |

Table of Contents

1. Introduction 6

1.1 Purpose 6

1.2 Scope 6

1.3 Definitions, Acronyms and Abbreviations 6

1.4 References 6

1.5 Overview 6

2. Overall Description 7

2.1 Use-Case Model Survey 7

2.1.1 Introduction 7

2.1.2 Survey Description 7

2.1.3 Use-Case Model Hierarchy 7

2.1.4 Diagrams of the Use-Case Model 8

2.2 Assumptions and Dependencies 8

3. Requirements 8

3.1 Use-Case Specifications 8

3.2 Functionality

3.3 Usability

3.4 Reliability 9

3.5 Performance

3.6 Supportability

3.7 Design Constraints 9

3.8 Online User Documentation and Help System Requirements 9

3.9 Purchased Components 10

3.10 Interfaces 10

3.10.1 User Interfaces 10

3.10.2 Hardware Interfaces 10

3.10.3 Software Interfaces 10

3.10.4 Communications Interfaces 10

3.11 Licensing Requirements 10

3.12 Legal, Copyright and Other Notices 10

3.13 Applicable Standards 10

Modern Software Requirements Specification

# Introduction

This SRS contains the documentation of the requirements used in creating the Online Course Reservation System. It also contains the Use case model hierarchy and the diagrams of the use case model. The software requirement specification is mainly used for requirement elicitation .The users of SRS are the client, the users, the system analysts and the system designers.

## Purpose

The main purpose of SRS document is to illustrate the user requirement of the stock maintenance system. This document is developed after a number of consultations with the client. This document will also act as a contract between the client and the developer in case of any dispute during the delivery.

## Scope

The scope of the Online Course Reservation System is :

1. A student should be able to

* Login to the software through the first page of the application
* Enter the details clearly
* Enter the courses information clearly

2. The administrator should recognize the personal and account details

3. The course requested by the student should be reserved.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| VB | Visual Basic, Front end |
| OCRS | Online Course Reservation System |

## References

IEEE SRS Format

Rational Application Developer Tutorial, IBM Publications

http://publib.boulder.ibm.com/

http://www.iitd.ernet.in/jee/

http://www.aieee.nic.in/

http://gate.iitm.ac.in/

## Overview

The Online Course Reservation System helps a student to reserve a course from a college of his or her choice online. Based on the qualifications of the student the system determines if the student is eligible for the course he has opted for. Next, student can proceed to the payment step, where the course fee can be paid online.

# Overall Description

This section describes the functions of the project and their aims. It also includes the constraints and the requirements of the project. The Online Course Reservation System is aimed at helping student to reserve a course in a college of their preference. The perspective of project is to efficiently manage reservation of seats online in a simple and convenient manner. The product has a simple and easy user interface.

## Use-Case Model Survey

* Login
* Registration for new user
* Enter students details
* Check student information
* Select the college
* Select the course
* Online fees payment
* Course reservation

### Introduction

It is a graph of actors, a set of use cases enclosed by a system boundary and the association between actors and the use cases. A use case is a special flow of event that occurs in the system. An actor is a user playing a role with respect to the system.

### Survey Description

# Login- Existing user can enter in to the account

# Registration for new user- New user to create their account

# Enter student details- Students will give academic certificates

# Check student information- Administrator will check eligibility of the student

# Select college- The user will select the college

# Select the course- Students will select the course

# Online fees payment- Online fees payment will be made using credit/debit card or net banking.

### Use-Case Model Hierarchy

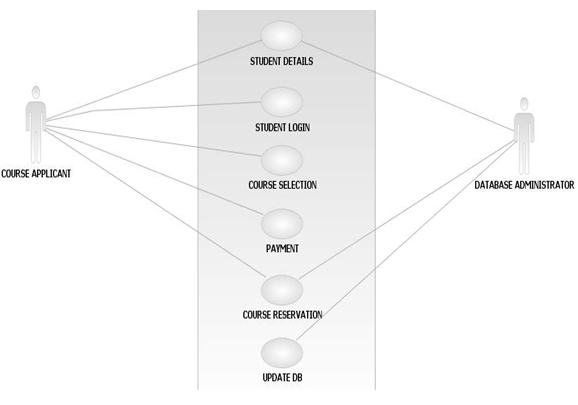
Following are the actors in the in the system:

**Student**- One who wants to Register through online and fill the form for writing the exam

**Administrator**- One who get the personal details and check the eligibility of the candidate.

### Diagrams of the Use-Case Model

Diagrams, primarily use-case diagrams, of the entire use case model are included here



## Assumptions and Dependencies

* The user must have connected to the internet to use the system
* The user’s computer must be Windows 98 or later version platform
* TCP/IP protocol must be installed to communicate through HTTP message
* The accuracy of the information of users is the responsibility of all users

# Requirements

Online Course Reservation System requires Visual Basic 6.0 and Oracle 10g to be installed in the system. Also there will be a database which will keep all the records done by user while registration and course selection.

## Use-Case Specifications

The various use cases are

* Student details
* Student Login
* Course Selection
* Payment
* Course reservation
* Update database

## Functionality

The main functionality of system is that we can get from online even at our home itself.

## Usability

Online Course Reservation System is very user friendly. Users can easily register and reserve a course of their choice.

## Reliability

The system shall operate 95% of the time. The number of defect should not exceed 10 per function. In addition, before the submission of the final release the calendar must be tested in case of the defects over 10 per function.

## Performance

* The system shall accommodate 500 users during the peak usage time window of 8:00am to 10:00am local time, with estimated average session duration of 45
* All web pages generated by the system shall be fully downloadable in not more than 10 seconds over a 40KBps modem connection.
* The system can display confirmation message to users within 4 seconds after the user submits information to the system.

## Supportability

The system is supported by Visual Basic for design and Database Management Concepts for backend database storage.

## Design Constraints

The design constraints of Hardware and Software

### The Hardware:

The various hardware constraints needed are CPU, Processor speed, Coprocessor speed etc.

### The software:

The various software constraints are Front end, Back end and OS.

## Online User Documentation and Help System Requirements

None

## Purchased Components

None

## Interfaces

It describes the various interface needed for the user

### User Interfaces

The user interface is Windows XP Operating System.

### Hardware Interfaces

The Hardware interfaces are:

CPU : Core 2 Duo

Processor speed : 2 GHZ

Coprocessor : Built in

Total RAM : 3 GB

Hard Disk : 250 GB

Keyboard : 105 Keys

Mouse : Logitech Mouse

Operating system : Windows XP/2007/Vista,

The software interfaces are:

Front end : VB

Back end : Oracle

Operating System : Windows XP

### Communications Interfaces

The communication interface is LAN (Local Area Interface)

## Licensing Requirements

The License should be obtained from cite ware which allows the modification of the source code at any time.

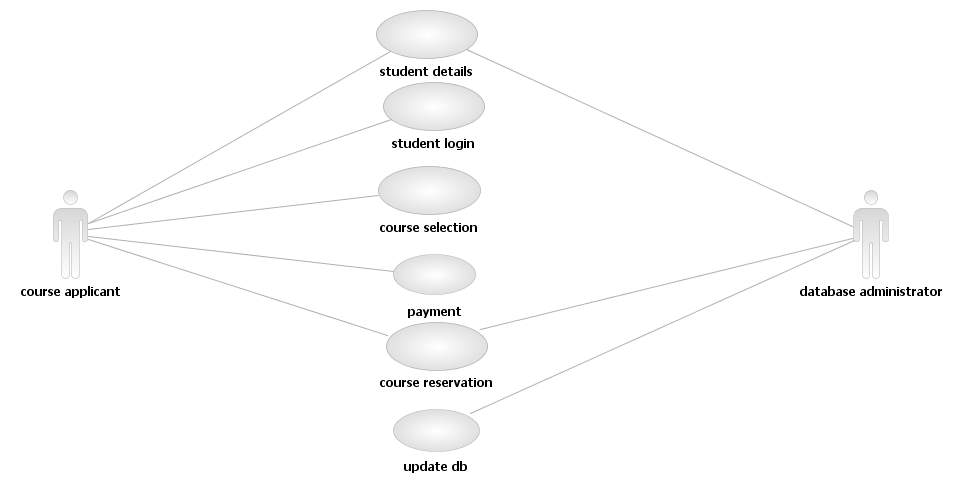
## Legal, Copyright and Other Notices

None

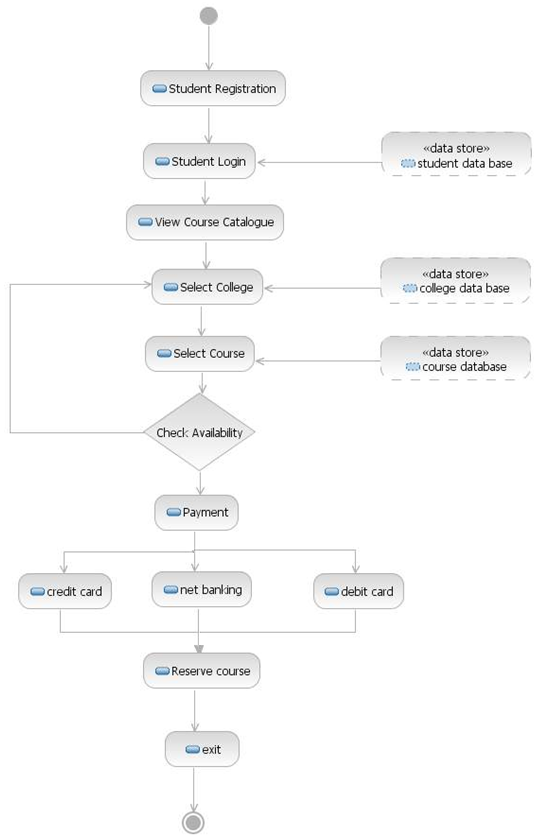
## Applicable Standards

None

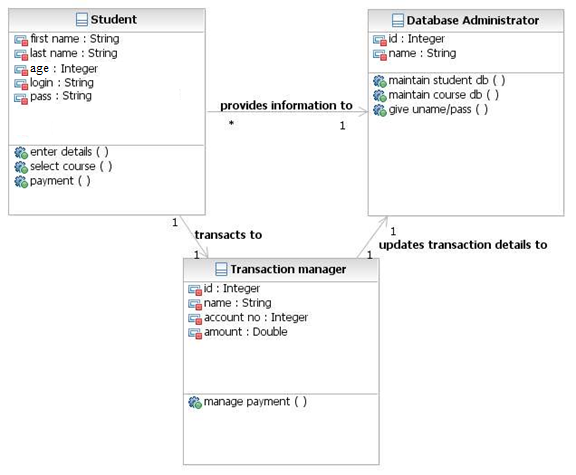
**Use Case Model**



**ACTIVITY DIAGRAM**



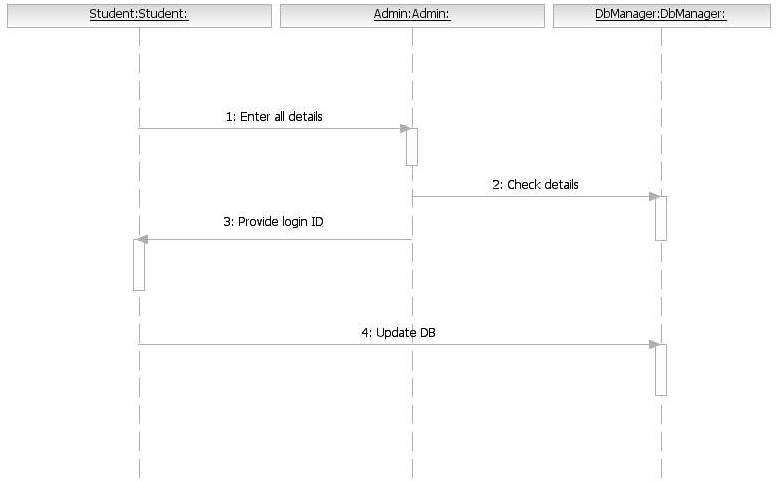
**CLASS DIAGRAM**



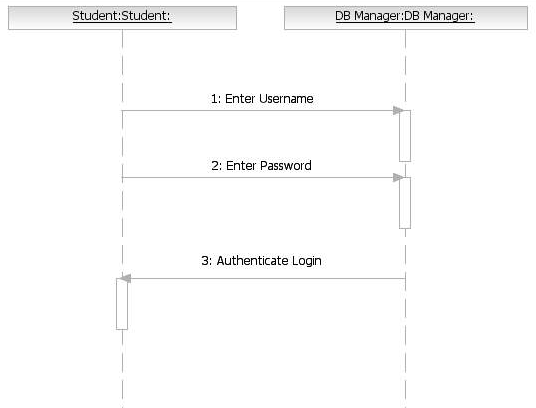
**INTERACTION DIAGRAMS**

Sequence Diagram

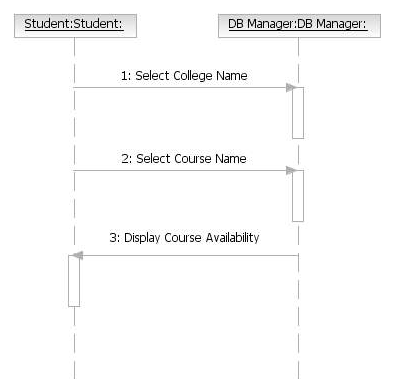
Student Registration



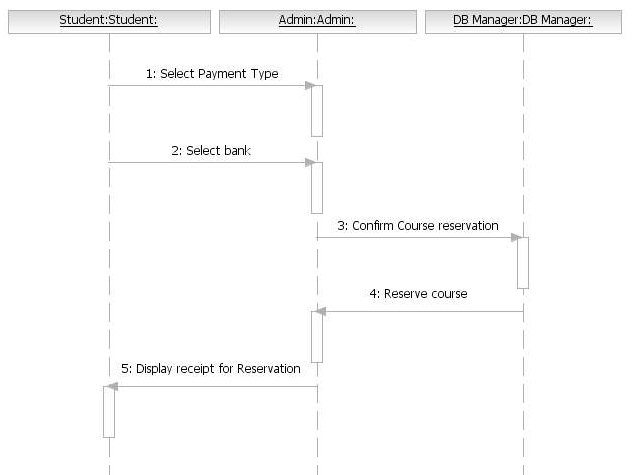
Student Login



Availability

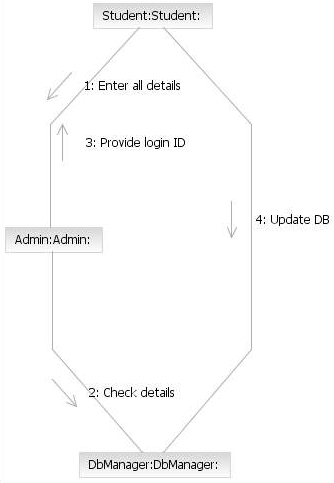


Payment and Confirmation

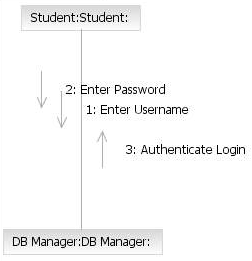


Collaboration Diagram

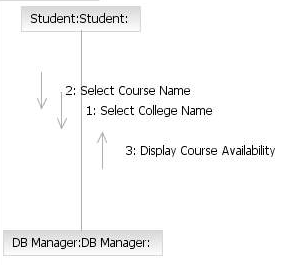
Student Registration



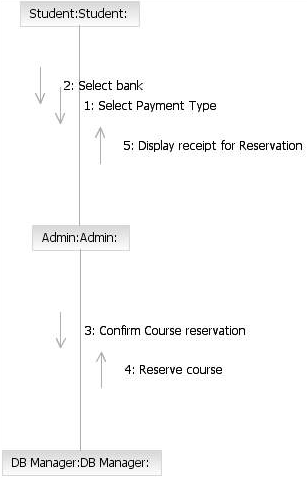
Student Login



Availability

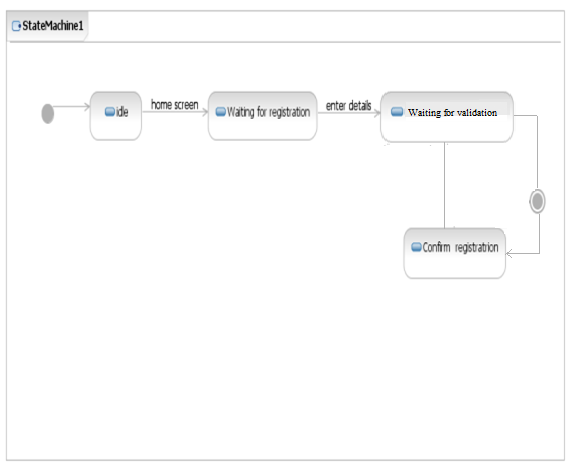


Payment and Confirmation

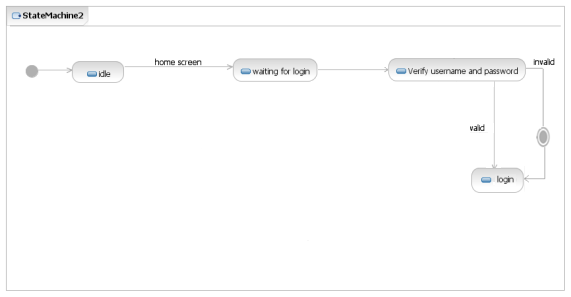


**STATE CHART DIAGRAM**

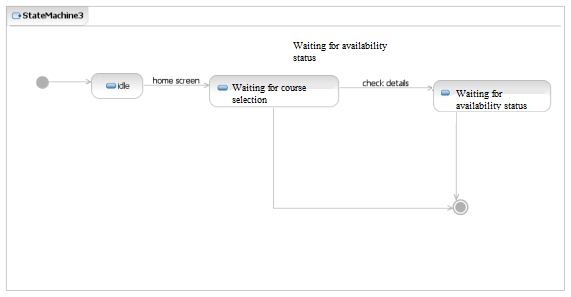
Student Registration



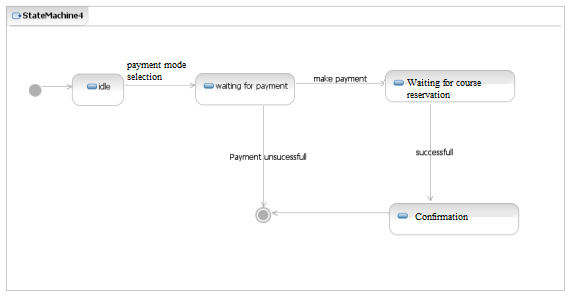
Login



Availability



Payment and Confirmation



**PACKAGE DIAGRAM**



.

**TECHNICAL SERVICES**

**SQL Queries:**

SQL> create table stud(fname varchar(20),lname varchar(20),age varchar(20),login varchar(20),password varchar(20),rollno number(20));

Table created.

SQL> create table avail(college varchar(20),cse number(3),ece number(3),mech number(3),eee number(3));

Table created.

SQL> create table login(login varchar(20),password varchar(20));

Table created.

SQL> create table course(course varchar(20),college varchar(20),cid number(5));

Table created.

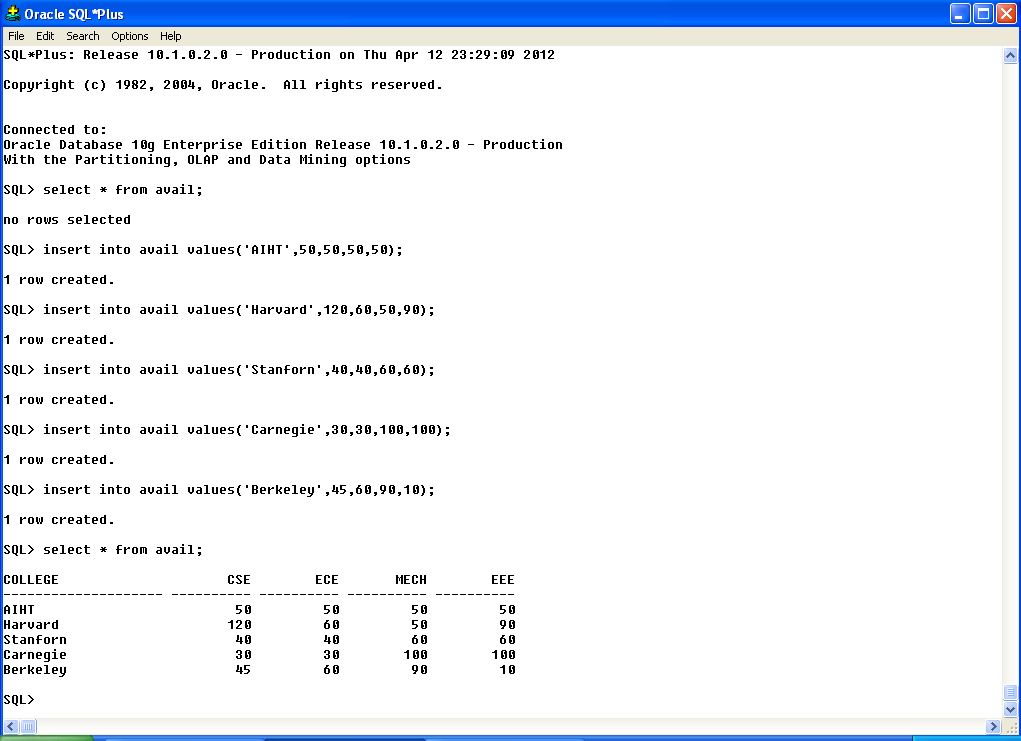
SQL> insert into avail values('AIHT',50,50,50,50);

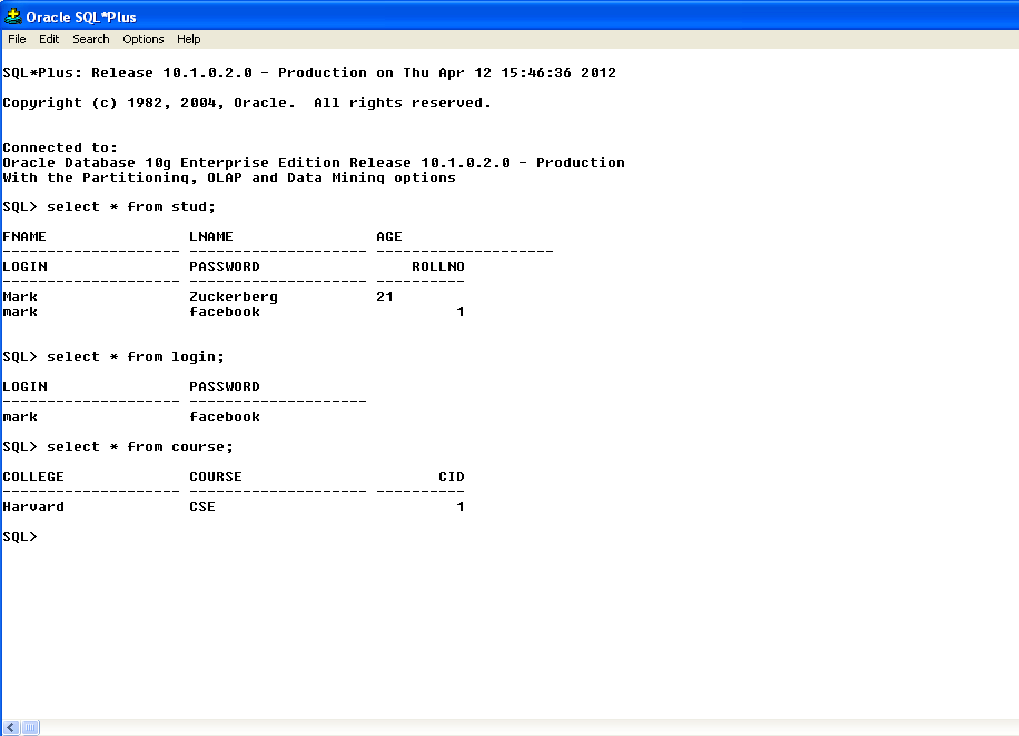
SQL> insert into avail values('Harvard',120,60,50,90);

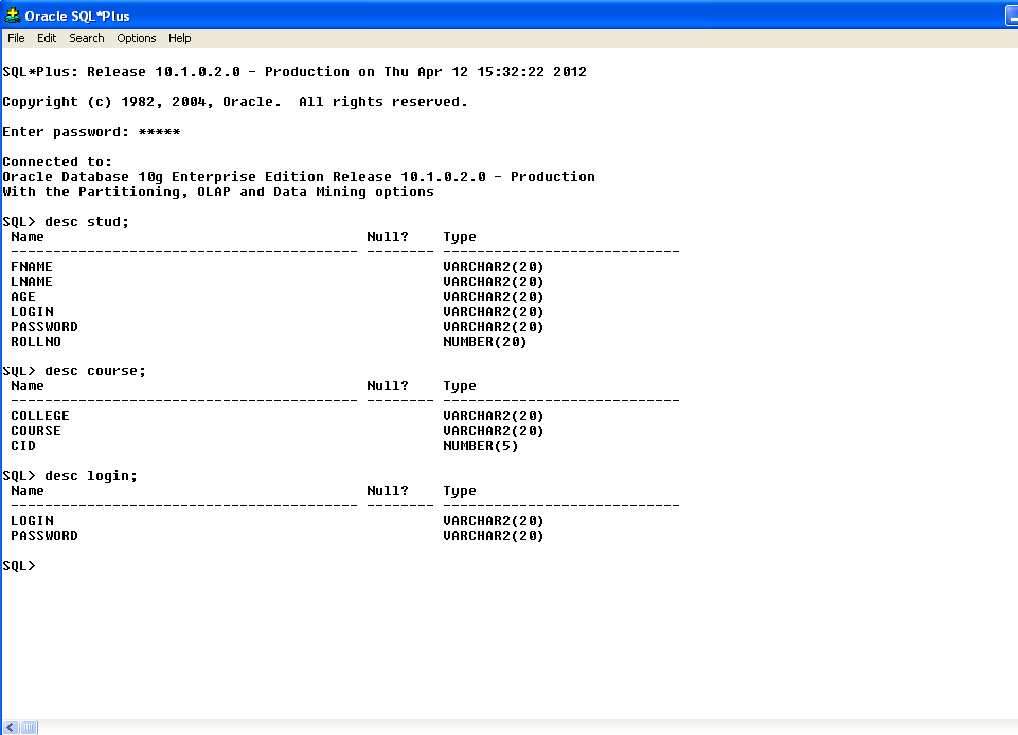
SQL> insert into avail values('Stanforn',40,40,60,60);

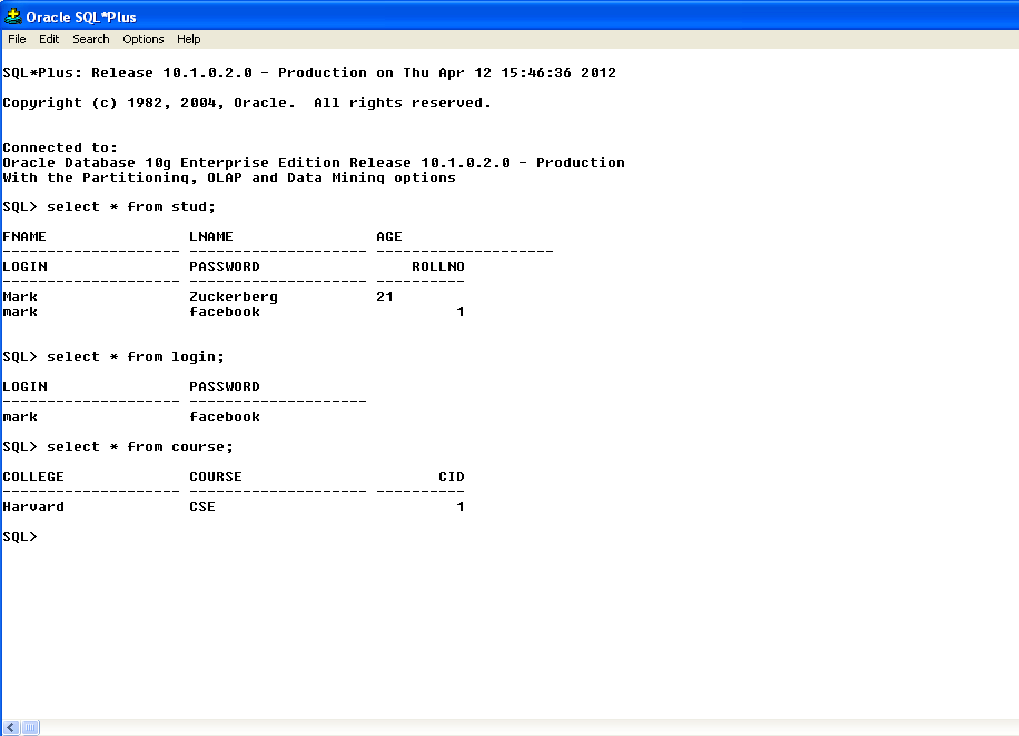
SQL> insert into avail values('Carnegie',30,30,100,100);

SQL> insert into avail values('Berkeley',45,60,90,10);









**DOMAIN OBJECT LAYER**

Administrator.java

public class administrator

{

private Integer id;

public Integer getId()

{

// begin-user-code

return id;

// end-user-code

}

public void setId(Integer theId)

{

// begin-user-code

id = theId;

// end-user-code

}

private String name;

public String getName()

{

// begin-user-code

return name;

// end-user-code

}

public void setName(String theName)

{

// begin-user-code

name = theName;

// end-user-code

}

public void maintainstudentdb()

{

// begin-user-code

// TODO Auto-generated method stub

// end-user-code

}

public void maintaincoursedb()

{

// begin-user-code

// TODO Auto-generated method stub

// end-user-code

}

public void giveunamepass()

{

// begin-user-code

// TODO Auto-generated method stub

// end-user-code

}

}

Student.java

public class student

{

private String firstname;

public String getFirstName()

{

// begin-user-code

return name;

// end-user-code

}

public void setFirstname(String theFirstname)

{

// begin-user-code

firstname = theFirstname;

// end-user-code

}

private String lastname;

public String getLastname()

{

// begin-user-code

return name;

// end-user-code

}

public void setLastname(String theLastname)

{

// begin-user-code

lastname = theLastname;

// end-user-code

}

private integer age;

public integer getAge()

{

// begin-user-code

return age;

// end-user-code

}

public void setAge(integer theAge)

{

// begin-user-code

age = theAge;

// end-user-code

}

private String login;

public String getLogin()

{

// begin-user-code

return login;

// end-user-code

}

public void setLogin(String theLogin)

{

// begin-user-code

login = theLogin;

// end-user-code

}

private String pass;

public String getPass()

{

// begin-user-code

return pass;

// end-user-code

}

public void setPass(String thePass)

{

// begin-user-code

pass = thePass;

// end-user-code

}

private administrator administrator;

public administrator getAdministrator()

{

// begin-user-code

return administrator;

// end-user-code

}

public void setAdministrator(administrator theAdministrator)

{

// begin-user-code

administrator = theAdministrator;

// end-user-code

}

private transactionmanager transactionmanager;

public transactionmanager getTransactionmanager()

{

// begin-user-code

return transactionmanager;

// end-user-code

}

public void setTransactionmanager(transactionmanager theTransactionmanager)

{

// begin-user-code

transactionmanager = theTransactionmanager;

// end-user-code

}

public void enterdetails()

{

// begin-user-code

// TODO Auto-generated method stub

// end-user-code

}

public void selectcourse()

{

// begin-user-code

// TODO Auto-generated method stub

// end-user-code

}

public void payment()

{

// begin-user-code

// TODO Auto-generated method stub

// end-user-code

}

}

Transactionmanage.java

public class transactionmanager

{

private integer id;

public integer getId()

{

// begin-user-code

return id;

// end-user-code

}

public void setId(integer theId)

{

// begin-user-code

id = theId;

// end-user-code

}

private String name;

public String getName()

{

// begin-user-code

return name;

// end-user-code

}

public void setName(String theName)

{

// begin-user-code

name = theName;

// end-user-code

}

private Integer accountno;

public Integer getAccountno()

{

// begin-user-code

return accountno;

// end-user-code

}

public void setAccountno(Integer theAccountno)

{

// begin-user-code

accountno = theAccountno;

// end-user-code

}

private Double amount;

public Double getAmount()

{

// begin-user-code

return amount;

// end-user-code

}

public void setAmount(String theAmount)

{

// begin-user-code

amount = theAmount;

// end-user-code

}

private databaseadministrator databaseadministrator;

public databaseadministrator getDatabaseadministrator()

{

// begin-user-code

return databaseadministrator;

// end-user-code

}

public void setDatabaseadministrator(administrator theDatabaseadministrator)

{

// begin-user-code

databaseadministrator = theDatabaseadministrator;

// end-user-code

}

public void managepayment()

{

// begin-user-code

// TODO Auto-generated method stub

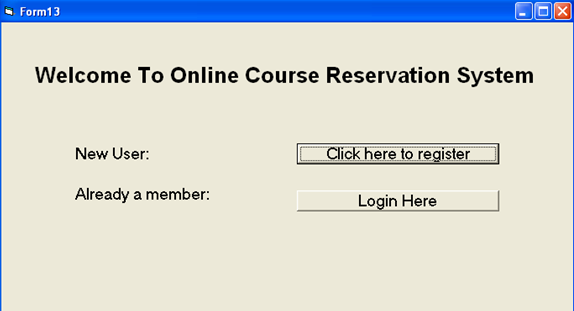
// end-user-code

}

}

**USER INTERFACE LAYER**

**Welcome Screen**



Private Sub Command1\_Click()

Form1.Show

Form13.Hide

End Sub

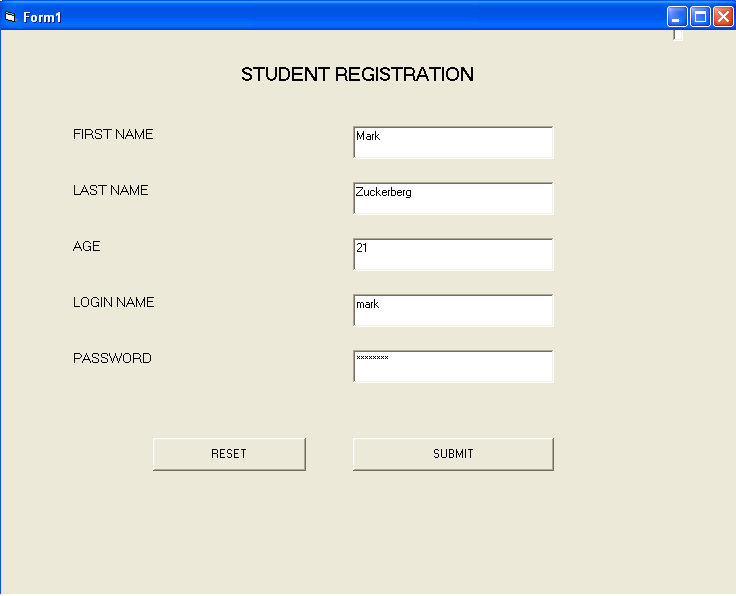
Private Sub Command2\_Click()

Form10.Show

Form13.Hide

End Sub

**Student Registeration**



Dim cn As New ADODB.Connection

Dim rs As New ADODB.Recordset

Dim rs1 As New ADODB.Recordset

Dim roll As Integer

Private Sub Command1\_Click()

rs.AddNew

roll = roll + 1

rs.Fields(0) = Text1.Text

rs.Fields(1) = Text2.Text

rs.Fields(2) = Text3.Text

rs.Fields(3) = Text4.Text

rs.Fields(4) = Text5.Text

rs.Fields(5) = roll + rs.Fields(5).Value

rs.Update

rs1.AddNew

rs1.Fields(0) = Text4.Text

rs1.Fields(1) = Text5.Text

rs1.Update

MsgBox ("Registration Successful !!"), vbInformation

Form1.Hide

Form11.Show

End Sub

Private Sub Form\_Load()

cn.Open "ocrs", "scott", "tiger"

rs.Open "select \* from stud", cn, adOpenDynamic, adLockOptimistic

rs1.Open "select \* from login", cn, adOpenDynamic, adLockOptimistic

MsgBox ("Connected to The Database")

'If conn.State = 0 Then

'conn.Open "new"

'End If

End Sub

**Student Login**



Dim rs As New ADODB.Recordset

Dim cn As New ADODB.Connection

Dim Msg As String

Private Sub Command1\_Click()

Do Until rs.EOF

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

MsgBox ("Login Successful !")

Form10.Hide

Form12.Show

Exit Sub

Else

rs.MoveNext

End If

Loop

Msg = MsgBox("Invalid password, try again!", vbOKCancel)

rs.MoveFirst

If (Msg = 1) Then

Form10.Show

Else

End

End If

End Sub

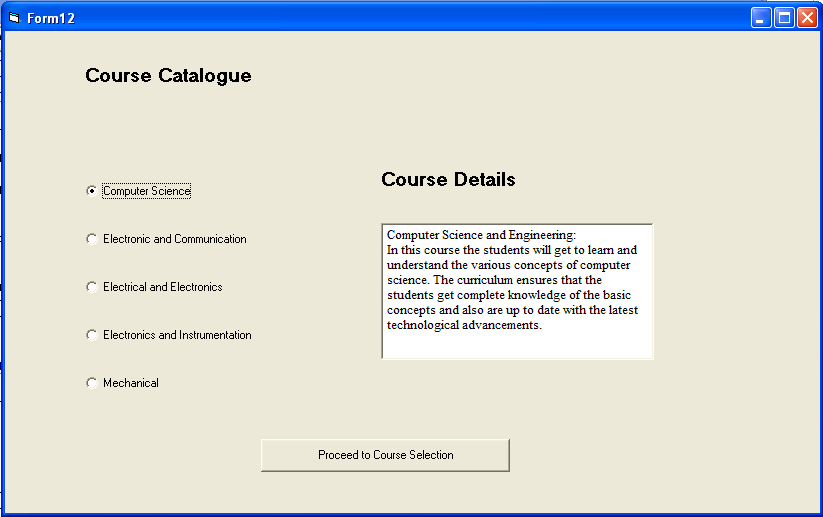
Private Sub Form\_Load()

cn.Open "ocrs", "scott", "tiger"

rs.Open "select \* from login", cn, adOpenDynamic, adLockOptimistic

End Sub

**Course Catalogue**



Private Sub Command1\_Click()

Form7.Show

Form12.Hide

End Sub

Private Sub Option1\_Click()

If Option1.Enabled = True Then

Text2.Text = "Computer Science and Engineering: In this course student will get to learrn about computer....."

End If

End Sub

Private Sub Option2\_Click()

If Option2.Enabled = True Then

Text2.Text = "Electronics and Communication Engineering:In this course student will get to learrn about Electronics....."

End If

End Sub

Private Sub Option3\_Click()

If Option3.Enabled = True Then

Text2.Text = "Electrical and Electronics Engineering:In this course student will get to learrn about Electrical....."

End If

End Sub

Private Sub Option4\_Click()

If Option4.Enabled = True Then

Text2.Text = "Electronics and Instrumentation Engineering: In this course student will get to learrn about Instrumentation....."

End If

End Sub

Private Sub Option5\_Click()

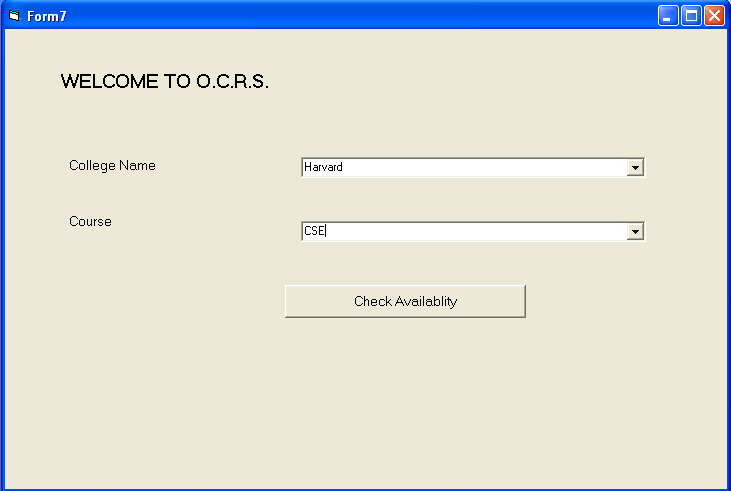
If Option5.Enabled = True Then

Text2.Text = "Mechanical Engineering : In this course student will get to learrn about Mechanical....."

End If

End Sub

**Course Selection**



Dim rs As New ADODB.Recordset

Dim rs1 As New ADODB.Recordset

Dim cn As New ADODB.Connection

Dim cid As Integer

Private Sub Command1\_Click()

rs.AddNew

rs.MoveFirst

cid = cid + 1

rs.Fields(0) = Combo4.Text

rs.Fields(1) = Combo5.Text

rs.Fields(2) = cid

If Combo4.Text = "AIHT" And Combo5.Text = "CSE" Then

rs.Fields(3) = rs1.Fields(1).Value

Else

If Combo4.Text = "AIHT" And Combo5.Text = "ECE" Then

rs.Fields(3) = rs1.Fields(2).Value

Else

If Combo4.Text = "AIHT" And Combo5.Text = "MECH" Then

rs.Fields(3) = rs1.Fields(3).Value

Else

If Combo4.Text = "AIHT" And Combo5.Text = "EEE" Then

rs.Fields(3) = rs1.Fields(4).Value

Else

If Combo4.Text = "Harvard" And Combo5.Text = "CSE" Then

rs.Fields(3) = rs1.Fields(1).Value

Else

If Combo4.Text = "Harvard" And Combo5.Text = "ECE" Then

rs.Fields(3) = rs1.Fields(2).Value

Else

If Combo4.Text = "Harvard" And Combo5.Text = "MECH" Then

rs.Fields(3) = rs1.Fields(3).Value

Else

If Combo4.Text = "Harvard" And Combo5.Text = "EEE" Then

rs.Fields(3) = rs1.Fields(4).Value

Else

If Combo4.Text = "Stanford" And Combo5.Text = "CSE" Then

rs.Fields(3) = rs1.Fields(1).Value

Else

If Combo4.Text = "Stanford" And Combo5.Text = "ECE" Then

rs.Fields(3) = rs1.Fields(2).Value

Else

If Combo4.Text = "Stanford" And Combo5.Text = "MECH" Then

rs.Fields(3) = rs1.Fields(3).Value

Else

If Combo4.Text = "Stanford" And Combo5.Text = "EEE" Then

rs.Fields(3) = rs1.Fields(4).Value

Else

If Combo4.Text = "Carnegie" And Combo5.Text = "CSE" Then

rs.Fields(3) = rs1.Fields(1).Value

Else

If Combo4.Text = "Carnegie" And Combo5.Text = "ECE" Then

rs.Fields(3) = rs1.Fields(2).Value

Else

If Combo4.Text = "Carnegie" And Combo5.Text = "MECH" Then

rs.Fields(3) = rs1.Fields(3).Value

Else

If Combo4.Text = "Carnegie" And Combo5.Text = "EEE" Then

rs.Fields(3) = rs1.Fields(4).Value

Else

If Combo4.Text = "Berkeley" And Combo5.Text = "CSE" Then

rs.Fields(3) = rs1.Fields(1).Value

Else

If Combo4.Text = "Berkeley" And Combo5.Text = "ECE" Then

rs.Fields(3) = rs1.Fields(2).Value

Else

If Combo4.Text = "Berkeley" And Combo5.Text = "MECH" Then

rs.Fields(3) = rs1.Fields(3).Value

Else

If Combo4.Text = "Berkeley" And Combo5.Text = "EEE" Then

rs.Fields(3) = rs1.Fields(4).Value

Else: MsgBox "Invalid Selection"

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

rs.Update

MsgBox ("Course Available !")

Form8.Show

Form7.Hide

End Sub

Private Sub Form\_Load()

cn.Open "ocrs", "scott", "tiger"

rs.Open "select \* from course", cn, adOpenDynamic, adLockOptimistic

rs1.Open "select \* from avail", cn, adOpenDynamic, adLockOptimistic

Combo4.AddItem ("AIHT")

Combo4.AddItem ("Harvard")

Combo4.AddItem ("Stanford")

Combo4.AddItem ("Carnegie")

Combo4.AddItem ("Berkeley")

Combo5.AddItem ("CSE")

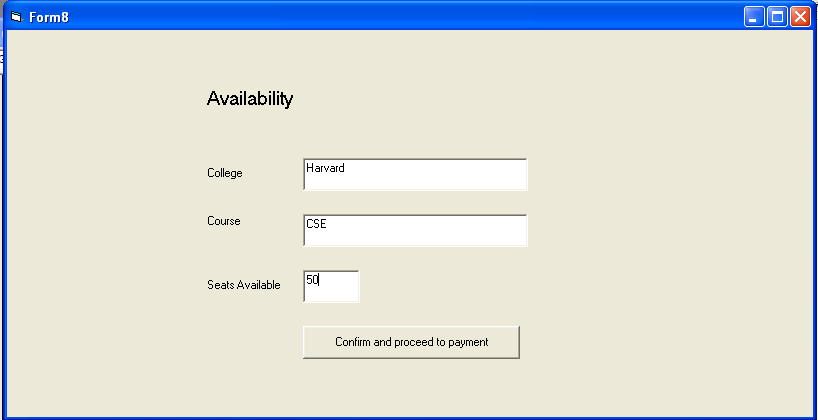
Combo5.AddItem ("ECE")

Combo5.AddItem ("MECH")

Combo5.AddItem ("EEE")

End Sub

**Availability**



Dim rs As New ADODB.Recordset

Dim rs1 As New ADODB.Recordset

Dim cn As New ADODB.Connection

Private Sub Command1\_Click()

rs.MoveFirst

If Text1.Text = "AIHT" And Text2.Text = "CSE" Then

rs.Fields(1).Value = rs.Fields(1).Value - 1

GoTo x

Else

If Text1.Text = "AIHT" And Text2.Text = "ECE" Then

rs.Fields(2).Value = rs.Fields(2).Value - 1

GoTo x

Else

If Text1.Text = "AIHT" And Text2.Text = "MECH" Then

rs.Fields(3).Value = rs.Fields(3).Value - 1

Else

If Text1.Text = "AIHT" And Text2.Text = "EEE" Then

rs.Fields(4).Value = rs.Fields(4).Value - 1

GoTo x

Else

If Text1.Text = "Harvard" And Text2.Text = "CSE" Then

rs.Fields(1).Value = rs.Fields(1).Value - 1

GoTo x

Else

If Text1.Text = "Harvard" And Text2.Text = "ECE" Then

rs.Fields(2).Value = rs.Fields(2).Value - 1

GoTo x

Else

If Text1.Text = "Harvard" And Text2.Text = "MECH" Then

rs.Fields(3).Value = rs.Fields(3).Value - 1

Else

If Text1.Text = "Harvard" And Text2.Text = "EEE" Then

rs.Fields(4).Value = rs.Fields(4).Value - 1

GoTo x

Else

If Text1.Text = "Stanford" And Text2.Text = "CSE" Then

rs.Fields(1).Value = rs.Fields(1).Value - 1

GoTo x

Else

If Text1.Text = "Stanford" And Text2.Text = "ECE" Then

rs.Fields(2).Value = rs.Fields(2).Value - 1

GoTo x

Else

If Text1.Text = "Stanford" And Text2.Text = "MECH" Then

rs.Fields(3).Value = rs.Fields(3).Value - 1

Else

If Text1.Text = "Stanford" And Text2.Text = "EEE" Then

rs.Fields(4).Value = rs.Fields(4).Value - 1

GoTo x

Else

If Text1.Text = "Carnegie" And Text2.Text = "CSE" Then

rs.Fields(1).Value = rs.Fields(1).Value - 1

GoTo x

Else

If Text1.Text = "Carnegie" And Text2.Text = "ECE" Then

rs.Fields(2).Value = rs.Fields(2).Value - 1

GoTo x

Else

If Text1.Text = "Carnegie" And Text2.Text = "MECH" Then

rs.Fields(3).Value = rs.Fields(3).Value - 1

Else

If Text1.Text = "Carnegie" And Text2.Text = "EEE" Then

rs.Fields(4).Value = rs.Fields(4).Value - 1

GoTo x

Else

If Text1.Text = "Berkeley" And Text2.Text = "CSE" Then

rs.Fields(1).Value = rs.Fields(1).Value - 1

GoTo x

Else

If Text1.Text = "Berkeley" And Text2.Text = "ECE" Then

rs.Fields(2).Value = rs.Fields(2).Value - 1

GoTo x

Else

If Text1.Text = "Berkeley" And Text2.Text = "MECH" Then

rs.Fields(3).Value = rs.Fields(3).Value - 1

Else

If Text1.Text = "Berkeley" And Text2.Text = "EEE" Then

rs.Fields(4).Value = rs.Fields(4).Value - 1

GoTo x

Else

If Text3.Text = 0 Then

MsgBox "Course Unavailable"

Else

MsgBox "Unexpected Error"

x:

rs.Update

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

Form8.Hide

Form2.Show

End Sub

Private Sub Form\_Load()

cn.Open "ocrs", "scott", "tiger"

rs.Open "select \* from avail", cn, adOpenDynamic, adLockOptimistic

rs1.Open "select \* from course ", cn, adOpenDynamic, adLockOptimistic

rs1.MoveLast

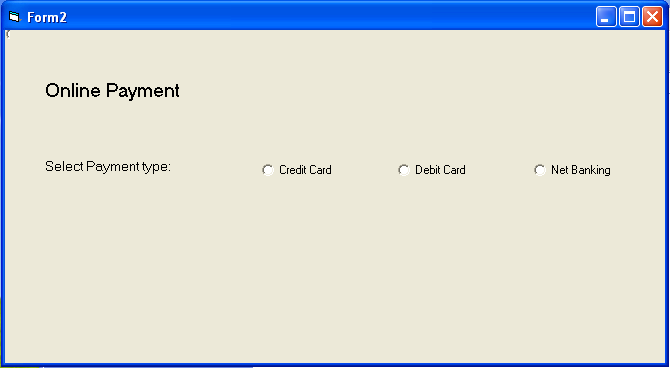
Text1.Text = rs1.Fields(0)

Text2.Text = rs1.Fields(1)

Text3.Text = rs1.Fields(3)

End Sub

**Payment**



Private Sub Option1\_Click()

Form2.Show

End Sub

Private Sub Option2\_Click()

Form4.Show

Form2.Hide

End Sub

Private Sub Option3\_Click()

Form5.Show

Form2.Hide

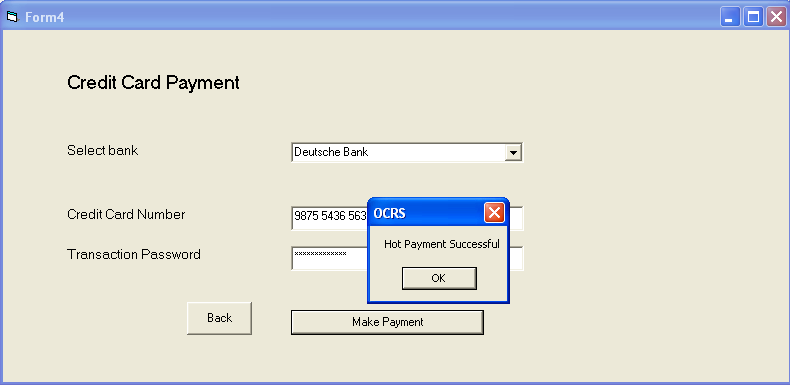
End Sub

Private Sub Option4\_Click()

Form3.Show

Form2.Hide

End Sub



Private Sub Command1\_Click()

MsgBox ("Hot Payment Successful")

Form9.Show

Form4.Hide

End Sub

Private Sub Command2\_Click()

Form2.Show

Form4.Hide

End Sub

Private Sub Form\_Load()

Combo8.AddItem ("BARCLAYS")

Combo8.AddItem ("Royal Bank of Scotland")

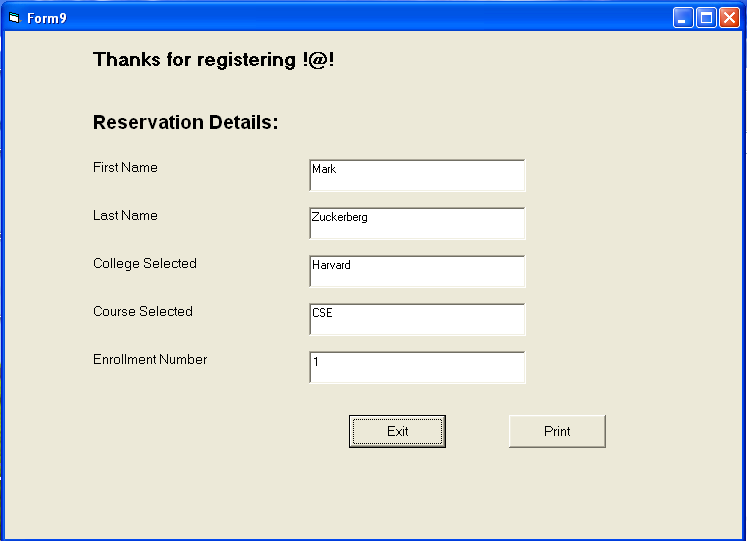
Combo8.AddItem ("Swiss Bank")

Combo8.AddItem ("CITI Bank")

Combo8.AddItem ("Deutsche Bank")

End Sub

**Reservation Details**



Dim rs As New ADODB.Recordset

Dim rs1 As New ADODB.Recordset

Dim cn As New ADODB.Connection

Private Sub Command1\_Click(Index As Integer)

Form9.Hide

End Sub

Private Sub Form\_Load()

cn.Open "ocrs", "scott", "tiger"

rs.Open "select \* from stud order by rollno", cn, adOpenDynamic, adLockOptimistic

rs1.Open "select \* from course order by cid", cn, adOpenDynamic, adLockOptimistic

rs.MoveLast

Text1.Text = rs.Fields(0)

Text2.Text = rs.Fields(1)

Text5.Text = rs.Fields(5)

rs1.MoveFirst

Text3.Text = rs1.Fields(0)

Text4.Text = rs1.Fields(1)

End Sub

**COMPONENT DIAGRAM**



**DEPLOYMENT DIAGRAM**

****

**RESULT**

Thus the Problem Statement, SRS document, UML Diagrams, Technical, Domain & Service layers for Online Course Reservation System was successfully developed.