

Bharatiya Vidya Bhavans Sardar Patel Institute of Technology

Munshi Nagar, Andheri (West), Mumbai 400 058

Real Time Surveillance Using Identity Based System

submitted by

MANISH M. D'SOUZA

SAINATH G. DUTKAR

Under the guidance of,

Prof. Sheetal Chaudhari



Department of Information Technology
Bharatiya Vidya Bhavans Sardar Patel Institute of Technology
Bhavans Campus, Munshi Nagar,
Andheri (West), Mumbai 400 058

UNIVERSITY OF MUMBAI 2013-14

CERTIFICATE

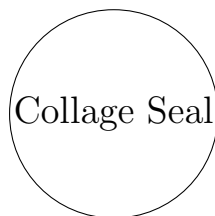
This is to certify that,
Manish Dsouza , Sainath Dutkar

Have satisfactorily carried out the project I entitled ”**Real Time Surveillance Using Identity Based System**” in partial fulfillment of Bachelor of Engineering in Information Technology, as laid down by University of Mumbai during the seventh semester academic year 2013-2014.

Internal Guide

Prof. Sheetal Chaudhari
Information Technology

Principal



HOD

Prof.Radha Shankarmani
Information Technology

.....
Internal Examiner

.....
External Examiner

Abstract

Traditionally, the task of surveillance video review has fallen to trained security personnel Capable of monitoring a modest number of incoming video streams. Such personnel become progressively less effective as the volume of video data grows and overloads the ability of the human eye/brain to process visual details. Exacerbating the problem is sheer fatigue, with long shift hours degrading the monitoring abilities of security staff still further. Adding more security personnel is a very costly option that, while reducing the number of video streams each security staff member must review, still fails to eliminate the inherent disadvantages (limited attention span, interruptions and distractions, fatigue) that accompany human monitoring of video data.

The purpose of this project is to design a surveillance system which would overcome this disadvantages of Traditional Systems and provide a better security solution. For this purpose we have taken the Domain as A school and provide parents live video surveillance of school classrooms. Each student's parents will be provided a unique user id and password using which they can access the live CCTV cam footage from the premises of school. Each student will be registered in the system. The students will be identified using the RFID tag attached to their school identity cards. RFID readers will be installed at the entrance of each block. The attendance table will be automatically updated and his parents will be notified. RFID reader will detect the student and the location of the student will be updated every time the student changes his location in the school. When the parents requests for the video feed, live video stream will be provided from the camera near the child. Since the video will be streamed only from camera near the child parents would not find the difficulty in finding their child. When the child leaves the school premises his parents will be informed about it. Our project will provide security as well as privacy to the parents. Also it will increase the access and will enable the parents to locate their children easily.

Acknowledgement

We have immense pleasure in presenting the synopsis report for our project entitled Real Time Surveillance Using Identity Based System. We would like to take this opportunity to express our gratitude to a number of people who have been sources of help and encouragement during the course of this project.

We are very grateful and indebted to our project guide Prof. Sheetal Chaudhari, for providing her enduring patience, guidance and invaluable suggestions. She was the one who never let our moral down and always supported us through our thick and thin. She is a constant source of inspiration for us and took utmost interest in our project.

We would also like to thank all the Staff Members for their valuable cooperation and permitting us to work in the I.T. Lab.

We are also thankful to all our classmates for giving us their useful advice and immense cooperation. Their support made the working of this project very pleasant. Last but not the least we thank the Almighty for giving us the strength and courage during the development of the project

MANISH M. D'SOUZA
SAINATH G. DUTKAR

Contents

List of Figures	iii
List of Tables	iv
Glossary	iv
1 Introduction	1
1.1 Problem Definition	1
1.2 Project scope	1
1.3 Existing system	2
1.4 Proposed system	2
1.5 Assumptions and constraints	3
1.5.1 Business Constraints	3
1.5.2 Technical Constraints	3
1.5.3 Environmental Constraints	3
1.5.4 Budgetary Constraints	3
1.5.5 Functionality Constraints	3
1.6 System Requirements	4
1.6.1 Software Interface	4
1.6.2 Hardware Interface	4
2 Literature Review	5
3 Methodology	7
3.1 Analysis	7
3.1.1 Use case diagram and specifications	7
3.1.2 Activity diagrams	11
3.1.3 Sequence diagram	12
3.1.4 Collaboration Diagram	13
3.1.5 Class Diagram	13
3.1.6 State Chart Diagram	14
3.1.7 Component Diagram	14
3.1.8 Deployment Diagram	15
3.2 Design	15
3.2.1 Architecture diagram	15
3.2.2 ER Diagram	16

3.3	Implementation	17
4	Result and Discussion	19
4.1	Sample code for key scenarios	19
4.2	Test Cases	41
4.3	Result Metrics	41
4.4	Result snapshots	42
5	Conclusion and Future Work	46
5.1	Conclusion	46
5.2	Future Scope	46
5.3	Adding Motion Detection And Alarm Generation	46
	Bibliography	50

List of Figures

3.1	Use case diagram and specifications	7
3.2	User Access Activity Diagram	11
3.3	Sequence diagram	12
3.4	Collaboration Diagram	13
3.5	Class Diagram	13
3.6	State Chart Diagram	14
3.7	Component Diagram	14
3.8	Deployment Diagram	15
3.9	Architecture diagram	15
3.10	ER Diagram	16
4.1	Teacher Login Window	42
4.2	User Workspace	42
4.3	Create User Form	42
4.4	Student Entry Form	43
4.5	Attendance Updation Form	43
4.6	Parent Login Page	44
4.7	Attendance Report	44
4.8	Live Camera Feed	45
5.1	Motion Detection And Alarm Generation	48

List of Tables

1.1	Software Interface	4
1.2	Hardware Interface	4
4.1	Result Metrics	41

Chapter 1

Introduction

1.1 Problem Definition

Surveillance is the monitoring of the behavior, activities, or other changing information, usually of people for the purpose of influencing, managing, directing, or protecting them. This can include observation from a distance by means of electronic equipment (such as CCTV cameras), or interception of electronically transmitted information (such as Internet traffic or phone calls); and it can include simple, relatively no- or low-technology methods such as human intelligence agents and postal interception. The word surveillance comes from a French phrase for "watching over" ("sur" means "from above" and "veiller" means "to watch"), and is in contrast to more recent developments such as surveillance.

Surveillance is very useful to colleges to maintain social control, recognize and monitor threats, and prevent/investigate criminal activity. With the help of such technologies such as high speed surveillance computers, RFID software, and ip camera users can now possess an unprecedented ability to monitor the activities of their students.

1.2 Project scope

In this competitive business environment where we are asked to cut costs and do more with less, process improvement is essential to staying alive and meeting our strategic objectives as well as Security is an Important aspect. By making the Surveillance On line it facilitates to keep a security watch from anywhere around the globe. So the work of security is not rested solely on the hands of security personnel.

A good surveillance tool will provide an easy way to access the security utilities, coordinate tasks, and reduce human efforts. It will also provide the flexibility and agility to support a constantly changing environment. The Demand for security of children in schools and colleges has increased and hence it is important to bring an innovative way of security in these institutes.

By providing the video feeds on line enables the parents to keep a watch on their children from

anywhere.

1.3 Existing system

CCTV

CCTV is an expensive device. Their installation is complicated, they are easily visible, and thus motive of hiding and surveying is not achieved. Requires large storage space for recorded full motion video doesn't provide the alert systems eg: sms or alarm. Manual work for observation is required to catch the nuisance behaviour.

IP CAMERA

An IP Camera is a stand-alone device which allows you to view live, full motion. IP Cameras can be used for surveillance of both homes and businesses. With the ability only to record live video to a remote location, IP Cameras allow you to make sure your recorded video is safe by storing it at a location that only you can access.

DRAW BACKS OF EXISTING SYSTEM

- It is not convenient as it consumes more time.
- No effective centralized control.
- Doesn't provide complete security
- System in computer requires large storage space
- Hardware cost is more

1.4 Proposed system

The purpose of this project is to design a surveillance system which would overcome the disadvantages of Traditional Systems and provide a better security solution. It provides direct access to the video feed based on the identity of the person.

Searching for a particular person from a large set of CCTV feeds makes it really difficult. This is because we do not have the exact idea about the location of the person. When we have the location of a person we can directly access the video feed of that location thereby reducing time required to locate a person. Also it reduces the loading of unnecessary data for the user providing exact required data.

1.5 Assumptions and constraints

1.5.1 Business Constraints

- In our system there is no such choice to cancel the appointment, prior to actual meeting.
- There will not be any credit transactions in our system.

1.5.2 Technical Constraints

- User should have basic knowledge of internet.
- One should have ip address, internet access and RFID hardware in order to operate this system.

1.5.3 Environmental Constraints

- The development or operating environment is new, and no members are familiar with it.
- Key decision-makers are difficult to contact when issues arise.
- The project environment is new and the components have not yet been successfully integrated.
- The project depends upon the successful and timely completion of associated projects.

1.5.4 Budgetary Constraints

- Statistics used in preparing the estimates are unreliable.
- Outside consulting requirements cannot be accurately estimated.

1.5.5 Functionality Constraints

- The project depends upon receiving data from other, external applications.

1.6 System Requirements

1.6.1 Software Interface

Table 1.1: Software Interface

Developer Side:	
Operating System	Windows XP
Application System	IIS 5.4
Database	SQL Server 2005
Programming language	.Net
Development IDE	Visual Studio 2008
Client Side:	
Operating System	Any
Web Browser	Any
Server side:	
Operating system	Any
Application Server	IIS 5.1
DBMS	SQL Server 2005

1.6.2 Hardware Interface

Table 1.2: Hardware Interface

Developer side:		
Processor	RAM	Disk Space
Intel p4 or equivalent	512MB	2 GB
Client Side:		
Intel p4 or equivalent	512MB	1 GB
Server Side:		
Server Environment Capable Hardware	2 GB	As per the Size of the required Data base

Chapter 2

Literature Review

Conventional video surveillance systems can record what they see, but they cant make sense of what they are viewing. That duty is typically the responsibility of security staff members, who have watched their jobs become increasingly demanding as the average number of surveillance cameras deployed grows. As one industry expert notes, In the past, security personnel viewed one camera on a single monitor. Now it is not uncommon to find them looking at 20 cameras linked to a single display. After 20 minutes of surveying, the human attention to video detail degenerates to an unacceptable level and video surveillance becomes meaningless. Traditional video surveillance can no longer meet the increased demands of the industry. Of course, some surveillance systems employ cameras that utilize video motion detection, but depending on how and where they are deployed, such systems can generate frequent false alarms. Motion detection makes no distinction between falling leaves, a eaping cator an adventurous burglar.

Also finding a particular person becomes difficult since there are a lot of camera feeds shown on a single screen at the same time and a human eye can look at and analyze a single window at a time. This makes the proccess slow and complex

Business intelligence solutions have revolutionized the enterprise, converting a relentless deluge of data into actionable information that can shape strategies, improve processes and boost bottom lines. The core benefit of such solutions is compelling: maximize the value of raw (unstructured) data through rigorous analysis that reveals key trends and correlations hidden within that data.

With the advent of surveillance digital video recorder (SDVR) systems, security professionals now face a similar challenge. While SDVRs and IP-based network DVRs (NDVRs) enable unprecedented access to vast quantities of high-resolution video images, deriving maximum benefit from this wealth of raw surveillance data requires meticulous review and analysis, sometimes on a frame-by-frame basis.

Traditionally, the task of surveillance video review has fallen to trained security personnel. Capable of monitoring a modest number of incoming video streams, such personnel become pro-

gressively less effective as the volume of video data grows and overloads the ability of the human eye/brain to process visual details. Exacerbating the problem is sheer fatigue, with long shift hours degrading the monitoring abilities of security staff still further. Adding more security personnel is a very costly option that, while reducing the number of video streams each security staff member must review, still fails to eliminate the inherent disadvantages (limited attention span, interruptions and distractions, fatigue) that accompany human monitoring of video Data

As can be seen below, the value proposed systems is comprehensive, encompassing greater cost- effectiveness and higher-quality surveillance, as well as greater scalability:More Cost-Effective
Conventional video surveillance environments require security personnel to spend many hours watching live or recorded video to analyze/identify suspicious events. By contrast, our systems can scan many thousands of hours of video data without human intervention.

More Accurate A variety of tests have shown that humans lose anywhere from 50% to 90 % of their visual perceptibility after 20 minutes of continuous video monitoring. The more video streams a person is 19 required to monitor in given period, the sooner impairments in visual perception manifest themselves. By contrast, our systems are immune to the fatigue, distractions and memory lapses tha plague human beings. Since surveillance is identity based user gets what is required and it not which is unnecessary.

Chapter 3

Methodology

3.1 Analysis

3.1.1 Use case diagram and specifications

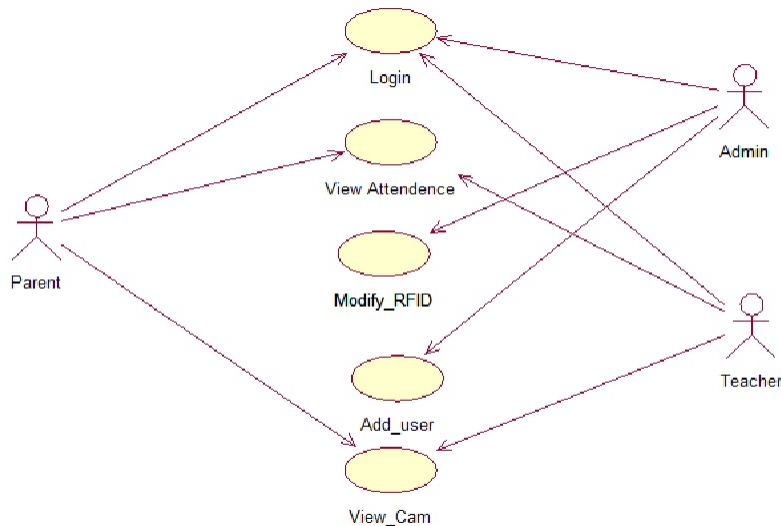


Figure 3.1: Use case diagram and specifications

It is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

The above Use Case Diagram defines the two users of the System and the various functions that they can perform like loading requirement sheet,saving the script ,etc.

Actor Documentation

Actor Name : Parent

Description : Any authorizes individual who wants to view details or feed

Actor Name : Admin

Description : Handling system level modification.

Actor Name: : Teacher

Description : Any authorizes teacher who wants to view details or feed

Use Case Documentation

- Use Case Name : Login
 - ID : 01
 - Importance Level : High
 - Primary Actor : Parent, Teacher, Admin
 - Usecase Type : Essential
 - Brief Description : When we are trying to gain access we need to first login to authenticate himself.
 - Trigger : Whenever the user begins a new session.
 - Normal Flow of event application. : 1.User opens the web browser or android application.
2.logs in.

●

- | | |
|-------------------------|--|
| Use Case Name | : viewaAttendance |
| ID | : 02 |
| Importance Level | : Medium |
| Primary Actor | : Parent, Teacher |
| Usecase Type | : Essential |
| Brief Description | : When user wants to view attendance details. |
| Trigger
attendance . | : Whenever user sends a request for showing |
| Normal Flow of event | : 1.User opens the web browser.
2.logs in.
3.View attendance |
- Use Case Name : Modify RFID
- | | |
|----------------------|---|
| ID | : 03 |
| Importance Level | : High |
| Primary Actor | : Admin |
| Usecase Type | : Essential |
| Brief Description | : When a new card is issued. |
| Trigger | : Whenever RFID modification is requested. |
| Normal Flow of event | : 1.User opens the web browser.
2.logs in.
3.modifies RFID string.
4.Maps it to database.
5.saves the database. |
-
- | | |
|---------------|-----------|
| Use Case Name | : AddUser |
| ID | : 04 |

Importance Level	: High
Primary Actor	: Admin
Usecase Typ	: Essential
Brief Description	: When a new user is to be added.
Trigger	: Whenever a new admission details come up .
Normal Flow of event	: 1.User opens the web browser. 2.logs in. 3.adds RFID string details. 4.Maps it to database. 5.saves the database.

•

Use Case Name	: View cam
ID	: 05
Importance Level	: High
Primary Actor	: Parent, Teacher
Usecase Type	: Essential
Brief Description	: When a user wants view live feed.
Trigger	: Whenever user requests for live feed .
Normal Flow of event	: 1.User opens the web browser. 2.logs in. 3.Requests for Cam feed. 4.UserId mapping with Reader location . 5.map location with camera IP 6. provide video feed

3.1.2 Activity diagrams

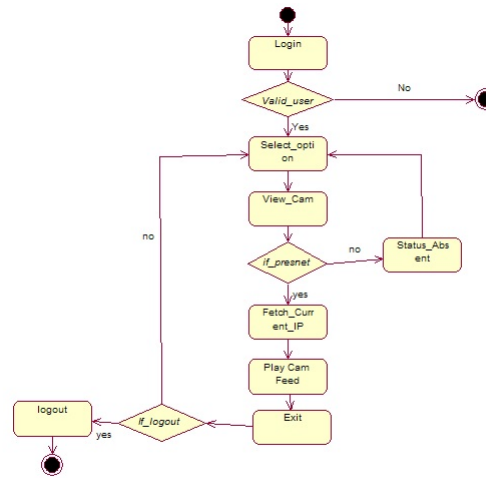


Figure 3.2: User Access Activity Diagram

3.1.3 Sequence diagram

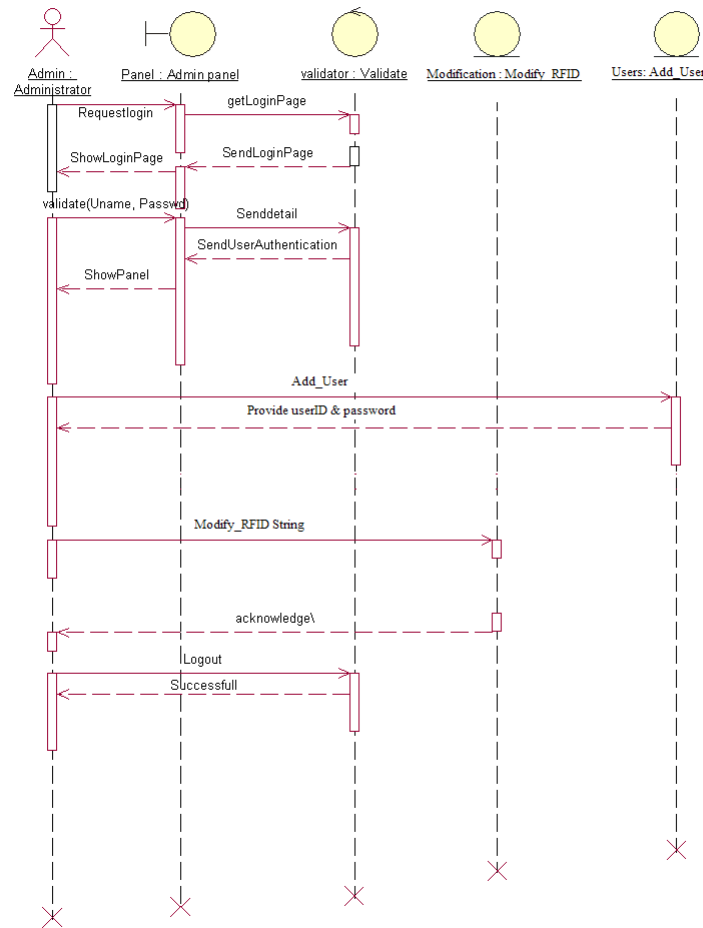


Figure 3.3: Sequence diagram

3.1.4 Collaboration Diagram

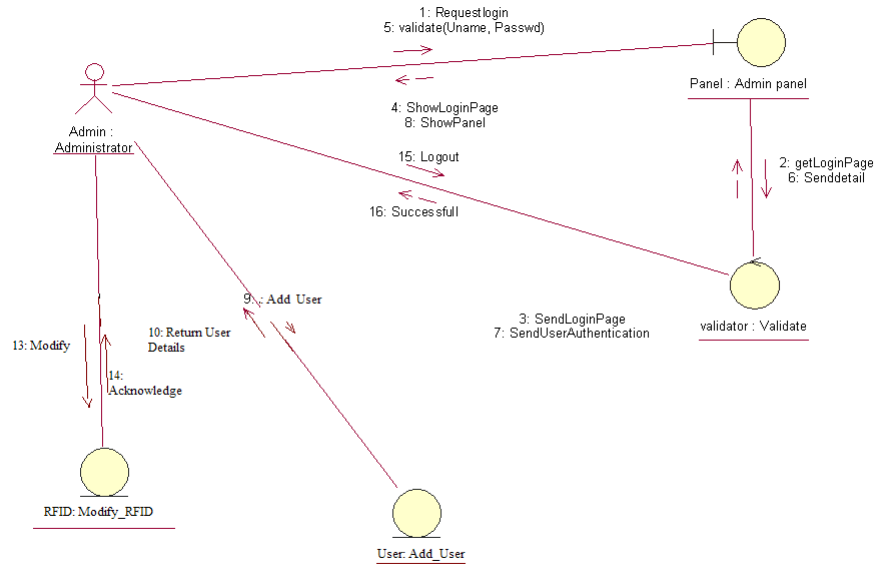


Figure 3.4: Collaboration Diagram

3.1.5 Class Diagram

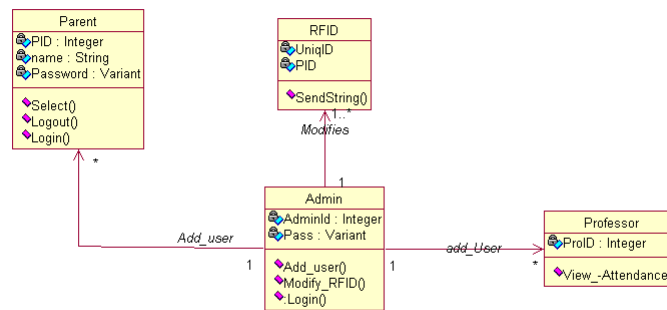


Figure 3.5: Class Diagram

3.1.6 State Chart Diagram

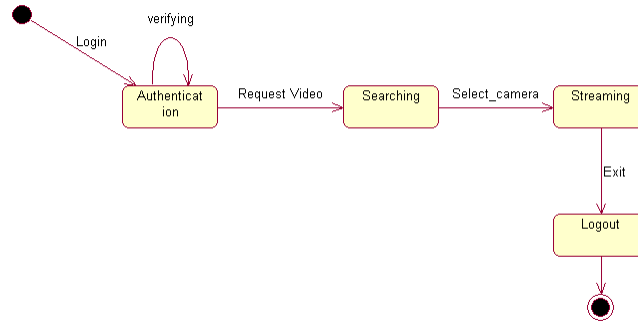


Figure 3.6: State Chart Diagram

3.1.7 Component Diagram

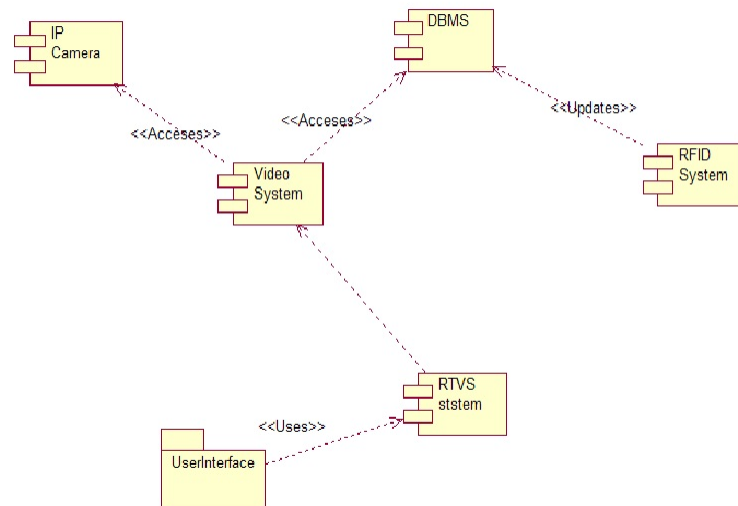


Figure 3.7: Component Diagram

3.1.8 Deployment Diagram

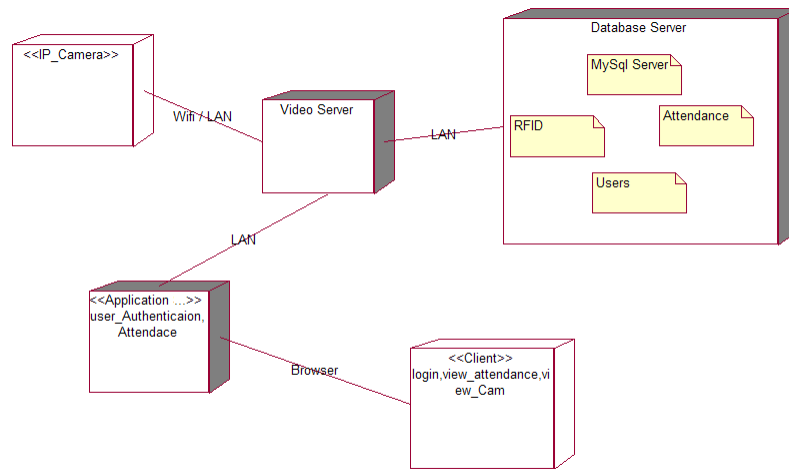


Figure 3.8: Deployment Diagram

3.2 Design

3.2.1 Architecture diagram

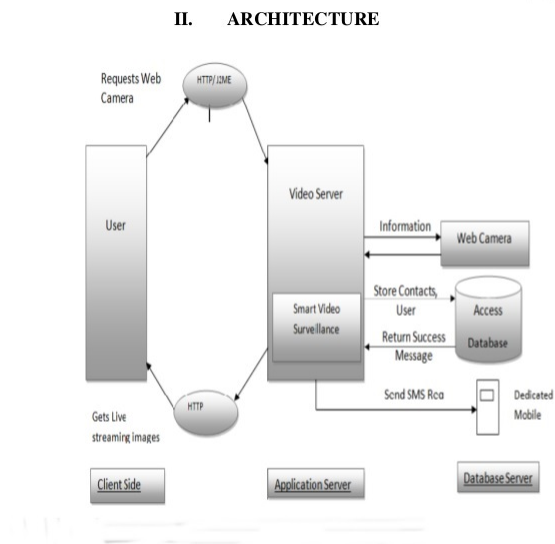


Figure 3.9: Architecture diagram

3.2.2 ER Diagram

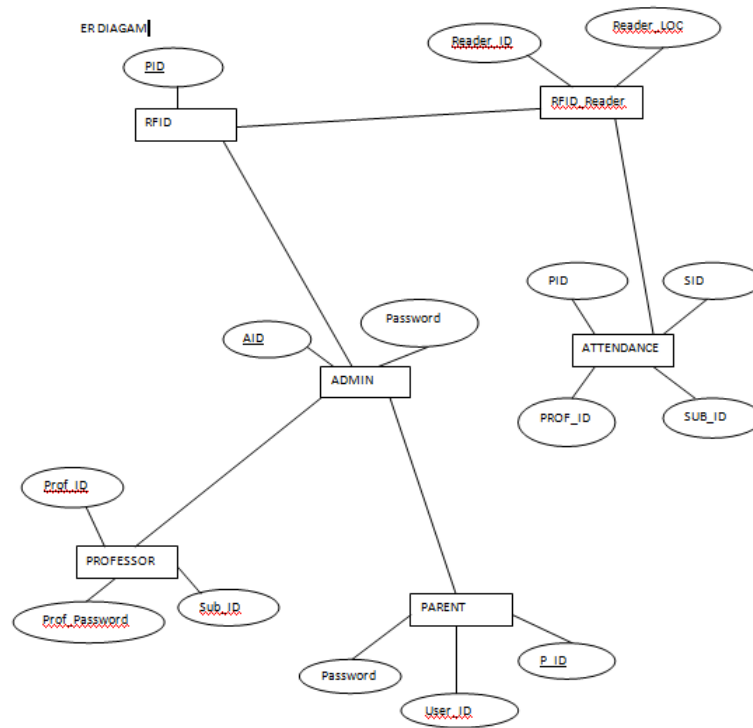


Figure 3.10: ER Diagram

3.3 Implementation

Algorithm: When student scans its RFID card the program does the following.

Step 01: It validates the RFID card.

```
strSQL = "select * from mstStudentEntry where RFID='" + txtRFID.Text.Trim() +
        """"
        dtRFID = ObjDataLayer.FillDataTable(strSQL)
        If dtRFID.Rows.Count > 0 Then step02
        Else
            Invalid user
        End if.
```

Step 02: If valid then it checks whether it is an entry transaction or exit transaction.

```
        strSQL = "select * from TransectionEntry where RFID='" +
            txtRFID.Text.Trim() + """"
        dtLogin = ObjDataLayer.FillDataTable(strSQL)
        If dtLogin.Rows.Count > 1 Then
            strSQL = "delete from TransectionEntry where RFID='" + txtRFID.Text.Trim()
                + """"
            ObjDataLayer.ExecuteNonQuery(strSQL)

        Else
            {
                strSQL = "Insert INTO TransectionEntry(TransectionID,CameraID,RFID
                    ,EntryDate) " & _
                    " Values('" + Trim(txtTransectionID.Text).Replace("'", "''") + " ', '"
                    + Trim(txtCameraID.Text).Replace("'", "''") + " ', '" +
                    Trim(txtRFID.Text.Trim()).Replace("'", "''") & " ', '" & Date.Now &
                    " ')"

                ObjDataLayer.ExecuteNonQuery(strSQL)
            }
        End If
```

After successful login The program operates as follows:

Step 03: The Parent Id is mapped with students Id from MasterStudentEntry Table.

```
"select te.AutoID,te.TransectionID [Transection ID],te.CameraID [Camera
    ID],te.RFID,se.StudentID [Student ID],se.StudentName [Student
    Name],se.Class,se.CollegeName [College Name]
    ,convert(varchar(50),te.EntryDate,103) [Entry Date] from TransectionEntry te
```

```
left join mstStudentEntry se on te.RFID=se.RFID "
```

Step 04 : Then we check whether student is present in transaction table using select query and RFID string obtained from above step.

```
"select * from mstStudentEntry where RFID='" + txtRFID.Text.Trim() + "'"
```

Step 05: If `true` then we select the camera Id present in Transaction table which denotes the `class` room of student.

```
"select te.AutoID,te.TransactionID [Transaction ID],te.CameraID [Camera ID],te.RFID,se.StudentID [Student ID],se.StudentName [Student Name],se.Class,se.CollegeName [College Name],convert(varchar(50),te.EntryDate,103) [Entry Date] from TransectionEntry te left join mstStudentEntry se on te.RFID=se.RFID where se.StudentID='" + txtStudentID.Text.Trim() + "' ";
```

```
dtTable = DataLayer.FillDataTable(strSQL);
```

```
strCamera = dtTable.Rows[0]["Camera ID"].ToString();
```

Step 06: Then we display video feed from the camera:

```
if (strCamera == "Camera1")
{
    iframe1.Attributes["src"] =
        "http://172.16.41.88:8080/browser.html";
    lblMessage.Text = "Camera 1";
}
else
{
    iframe1.Attributes["src"] =
        "http://192.168.1.8:8080/browser.html";
    lblMessage.Text = "Camera 2";
}
```

Chapter 4

Result and Discussion

4.1 Sample code for key scenarios

Student entry code:

```
Imports System.Text
```

```
Imports System.IO
```

```
Public Class frmStudentEntry
```

```
    '---serial port to listen to incoming data---
```

```
    Private WithEvents serialPort As New IO.Ports.SerialPort
```

```
    '---tag ID read from the reader---
```

```
    Private tagID As String = String.Empty
```

```
    '---the time that the tag ID was recorded---
```

```
    Private timeRecorded As DateTime = Now
```

```
    '---COM port to listen to---
```

```
    Const COM As String = "COM5"
```

```
    '---filename of the log file---
```

```
    Const FILE_NAME As String = "D:\Attendance.csv"
```

```
    '---the interval before the employee record is cleared
```

```
    ' from the screen (in seconds)---
```

```
    Const INTERVAL As Integer = 3
```

```
    Dim email As String
```

```
    Dim ObjDataLayer As New DataAccessLayer
```

```
    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
```

```
        System.EventArgs) Handles btnExit.Click
```

```
        Me.Close()
```

```
        If serialPort.IsOpen() Then
```

```
            serialPort.Close()
```

```

        serialPort.Dispose()
    End If
End Sub

Private Sub frmStudentEntry_Activated(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Me.Activated
    MDIForm.Panel1.SendToBack()
    MDIForm.Panel1.Dock = DockStyle.Fill
End Sub

Private Sub frmStudentEntry_FormClosing(ByVal sender As Object, ByVal e As
    System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    MDIForm.Panel1.BringToFront()
    MDIForm.Panel1.Dock = DockStyle.Fill

End Sub

Private Sub frmCreateUser_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    Try
        Me.ControlBox = False

        Timer1.Interval = INTERVAL * 1000 'convert to milliseconds
        '---open the serial port connecting to the reader---
        If serialPort.IsOpen Then
            serialPort.Close()
        End If
        Try
            With serialPort
                .PortName = COM
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
            End With
            serialPort.Open()
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        fillAutoID()
        FillGridView()
    
```

```

        ClearAll()
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try

End Sub

Private Sub DataReceived(ByVal sender As Object, ByVal e As
    System.IO.Ports.SerialDataReceivedEventArgs) Handles
    serialPort.DataReceived
    Try
        '---when incoming data is received, update the TagID
        ' textbox---
        txtRFID.BeginInvoke(New myDelegate(AddressOf updateTextBox), New
            Object() {})
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try

End Sub

'---update the Tag ID textbox---
Public Delegate Sub myDelegate()
Public Sub updateTextBox()
    '---for receiving plain ASCII text---
    If txtRfid.TextLength = 12 Then
        'txttagid.Text = ""
        With txtRfid
            .AppendText(serialPort.ReadExisting)
            .ScrollToCaret()
        End With
    Else
        With txtRfid
            .AppendText(serialPort.ReadExisting)
            .ScrollToCaret()
        End With
    End If

End Sub

Private Sub FillGridView()
    Try

```

```

Dim dtDetails As New DataTable
Dim strSQL As String = "select AutoID,StudentID [Student
    ID],StudentName [Student
    Name],RFID,EmailID1,Phone1,Class,CollegeName [College
    Name],Address,ParentUsername,ParentPassword,convert(varchar(50),EntryDate,1033)
    [Entry Date] from MstStudentEntry order by StudentName "
dtDetails = ObjDataLayer.FillDataTable(strSQL)
DgvUserData.DataSource = dtDetails
DgvUserData.Columns("AutoID").Visible = False
lblTotal.Text = "Total Count = " & dtDetails.Rows.Count
Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
End Try
End Sub
Private Sub ClearAll()
    Try

        txtStudentName.Text = ""
        txtRFID.Text = ""
        txtEmailID1.Text = ""
        txtPhone1.Text = ""
        txtClass.Text = ""
        txtCollegeName.Text = ""
        txtAddress.Text = ""
        btnUpdate.Enabled = False
        btnCreate.Enabled = True
        fillAutoID()
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try
End Sub
Private Sub fillAutoID()
    Try
        Dim dsDomestic As New DataTable
        dsDomestic = ObjDataLayer.FillDataTable("select
            Max(cast(substring(StudentID,12,50000) as int)) as StudentID from
            MstStudentEntry")

        If IsDBNull(dsDomestic.Rows(0)(0)) = True Then

```

```

        txtStudentID.Text = "RFIDS/Stu/-1"
    Else
        txtStudentID.Text = "RFIDS/Stu/-" & dsDomestic.Rows(0)(0) + 1
    End If

Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
End Try
End Sub

Private Function Validation() As Boolean
    Validation = True
    Try
        Dim strMsg As New StringBuilder

        If txtStudentName.Text = "" Then
            strMsg.AppendLine("Please enter Student Name")
            Validation = False
        End If

        If txtRFID.Text = "" Then
            strMsg.AppendLine("Please enter RFID")
            Validation = False
        End If

        If txtRFID.Text.Length <> 12 Then
            strMsg.AppendLine("Please enter Valid RFID")
            Validation = False
        End If

        If txtEmailID1.Text = "" Then
            strMsg.AppendLine("Please enter Email ID ")
            Validation = False
        End If

        If txtPhone1.Text = "" Then
            strMsg.AppendLine("Please enter phone ")
            Validation = False
        End If
    
```

```

    If txtClass.Text = "" Then
        strMsg.AppendLine("Please enter Class")
        Validation = False
    End If

    If txtCollegeName.Text = "" Then
        strMsg.AppendLine("Please enter College Name")
        Validation = False
    End If

    If txtAddress.Text = "" Then
        strMsg.AppendLine("Please enter Address")
        Validation = False
    End If

    If txtUsername.Text = "" Then
        strMsg.AppendLine("Please enter Username")
        Validation = False
    End If

    If txtPassword.Text = "" Then
        strMsg.AppendLine("Please enter Password")
        Validation = False
    End If

    If strMsg.Length > 0 Then
        MessageBox.Show(strMsg.ToString, strTitle, MessageBoxButtons.OK,
            MessageBoxIcon.Information, MessageBoxDefaultButton.Button1)
        Return False
    End If
    Return Validation
Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
End Try
End Function

Private Sub btnCreate_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreate.Click
    Try
        If Validation() = False Then

```



```

        Exit Sub
    End If
    Dim strSQL As String

    Dim m_SelectedStyle As New ComboBox()
    m_SelectedStyle.BackColor = Color.MediumAquamarine

    Dim dtUsername As New DataTable
    dtUsername = ObjDataLayer.FillDataTable("select StudentName from
        MstStudentEntry where StudentName Like '" + txtStudentName.Text +
        "';")
    If dtUsername.Rows.Count > 0 Then
        MessageBox.Show(txtStudentName.Text + " is already exists.",
            strTitle, MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
        Exit Sub
    End If

    txtStudentName.BackColor = Color.White
    'txtUsername.Text = ""
    If ObjDataLayer.ValidateEmailId(txtEmailID1) = False Then
        Exit Sub
    End If

    strSQL = "Insert INTO
        MstStudentEntry(StudentID,StudentName,RFID,EmailID1,Phone1,Class,CollegeName
        " & _
    " Values('" + Trim(txtStudentID.Text).Replace("'", "''") + "','" +
        Trim(txtStudentName.Text).Replace("'", "''") + "','" +
        Trim(txtRFID.Text.Trim()).Replace("'", "''") + "','" & _
    "'" + Trim(txtEmailID1.Text).Replace("'", "''") + "','" +
        Trim(txtPhone1.Text).Replace("'", "''") + "','" & _
    "'" + Trim(txtClass.Text).Replace("'", "''") + "','" +
        Trim(txtCollegeName.Text).Replace("'", "''") + "','" &
        txtAddress.Text.Trim.Replace("'", "''") + "','" + txtUsername.Text +
        "','" + txtPassword.Text + "','" & Date.Now & "')"
    ObjDataLayer.ExecuteNonQuery(strSQL)

    MessageBox.Show("Student Created Successfully..!", strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
    frmCreateUser_Load(sender, e)
Catch ex As Exception

```

```

        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try

End Sub

Private Sub DgvUserData_CellContentDoubleClick(ByVal sender As Object, ByVal
    e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
    DgvUserData.CellContentDoubleClick

End Sub

Private Sub fillData(ByVal StudentID As String)
    Try

        Dim dtData As New DataTable
        Dim strSQL As String = "select * from MstStudentEntry where
            StudentID='" & StudentID & "'"
        dtData = ObjDataLayer.FillDataTable(strSQL)
        txtStudentID.Text = IIf(IsDBNull(dtData.Rows(0)("StudentID")), "",
            dtData.Rows(0)("StudentID"))
        txtStudentName.Text = IIf(IsDBNull(dtData.Rows(0)("StudentName")),
            "", dtData.Rows(0)("StudentName"))
        txtRFID.Text = IIf(IsDBNull(dtData.Rows(0)("RFID")), "",
            dtData.Rows(0)("RFID"))
        txtEmailID1.Text = IIf(IsDBNull(dtData.Rows(0)("EmailID1")), "",
            dtData.Rows(0)("EmailID1"))
        txtPhone1.Text = IIf(IsDBNull(dtData.Rows(0)("Phone1")), "",
            dtData.Rows(0)("Phone1"))
        txtClass.Text = IIf(IsDBNull(dtData.Rows(0)("Class")), "",
            dtData.Rows(0)("Class"))
        txtCollegeName.Text = IIf(IsDBNull(dtData.Rows(0)("CollegeName")),
            "", dtData.Rows(0)("CollegeName"))
        txtAddress.Text = IIf(IsDBNull(dtData.Rows(0)("Address")), "",
            dtData.Rows(0)("Address"))
        txtUsername.Text = IIf(IsDBNull(dtData.Rows(0)("ParentUsername")),
            "", dtData.Rows(0)("ParentUsername"))
        txtPassword.Text = IIf(IsDBNull(dtData.Rows(0)("ParentPassword")),
            "", dtData.Rows(0)("ParentPassword"))
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try

```

End Sub

```
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles btnUpdate.Click
```

```
Try
```

```
    If Validation() = False Then
```

```
        Exit Sub
```

```
    End If
```

```
    If ObjDataLayer.ValidateEmailId(txtEmailID1) = False Then
```

```
        Exit Sub
```

```
    End If
```

```
    Dim strSQL As String
```

```
    strSQL = "update MstStudentEntry set StudentName='" &
```

```
        txtStudentName.Text.Trim.Replace("'", "") & "',RFID='" &
```

```
        txtRFID.Text.Trim.Replace("'", "") & "', EmailID1='" &
```

```
        txtEmailID1.Text.Trim.Replace("'", "") & "', " & _
```

```
    " Phone1='" & txtPhone1.Text.Trim.Replace("'", "") & "', " & _
```

```
    " Class='" & txtClass.Text.Trim.Replace("'", "") & "',CollegeName='"
```

```
        & txtCollegeName.Text.Trim.Replace("'", "") & "',Address='" &
```

```
        txtAddress.Text.Trim.Replace("'", "") & "' where StudentID='" &
```

```
        txtStudentID.Text.Trim & "'"
```

```
    ObjDataLayer.ExecuteNonQuery(strSQL)
```

```
    MessageBox.Show("Data Updated successfully !!", strTitle,
```

```
        MessageBoxButtons.OK, MessageBoxIcon.Information,
```

```
        MessageBoxDefaultButton.Button1)
```

```
    frmCreateUser_Load(sender, e)
```

```
Catch ex As Exception
```

```
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
```

```
        MessageBoxButtons.OK, MessageBoxIcon.Information,
```

```
        MessageBoxDefaultButton.Button1)
```

```
End Try
```

```
End Sub
```

```
Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles btnDelete.Click
```

```
Try
```

```
    If txtStudentName.Text = "" Then
```

```
        MessageBox.Show("Select a Student from Student Detail to delete",
```

```
            strTitle, MessageBoxButtons.OK, MessageBoxIcon.Information,
```

```
            MessageBoxDefaultButton.Button1)
```

```
        Exit Sub
```

```
    End If
```

```

    If MessageBox.Show("Are you sure, want to delete Student ? ",
        strTitle, MessageBoxButtons.YesNo, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1) = Windows.Forms.DialogResult.Yes
    Then
        Dim strSQL As String
        strSQL = "delete from MstStudentEntry where StudentId='" &
            txtStudentID.Text.Trim & "'"
        ObjDataLayer.ExecuteNonQuery(strSQL)
        MessageBox.Show("Data deleted successfully !!", strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End If

    frmCreateUser_Load(sender, e)
Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
End Try
End Sub

Private Sub DgvUserData_CellContentClick(ByVal sender As System.Object,
    ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
    DgvUserData.CellContentClick
    Try
        If e.ColumnIndex > -1 AndAlso e.RowIndex > -1 AndAlso TypeOf
            sender.CurrentCell Is DataGridViewTextBoxCell Then 'AndAlso TypeOf
            sender.CurrentCell Is DataGridViewTextBoxCell
            If sender.CurrentCell.EditedFormattedValue().ToString <> "" Then
                fillData(DgvUserData.CurrentRow.Cells("Student ID").Value)
                btnCreate.Enabled = False
                btnUpdate.Enabled = True
            End If
        End If
    End If

    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try
End Sub

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs)

```

End Sub

```
Private Sub btnClear_Click_1(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClear.Click
    ClearAll()
End Sub
```

```
Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs)
    frmCreateUser_Load(sender, e)
End Sub
```

```
Private Sub btnExport_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs)
    Me.Cursor = Cursors.WaitCursor
    Dim _excel As New Microsoft.Office.Interop.Excel.Application
    Dim wBook As Microsoft.Office.Interop.Excel.Workbook
    Dim wSheet As Microsoft.Office.Interop.Excel.Worksheet
    Try
```

```
        wBook = _excel.Workbooks.Add()
        wSheet = wBook.ActiveSheet()
        _excel.DisplayAlerts = False
        wSheet.Columns.AutoFit()
```

```
        Dim dt As System.Data.DataTable = DgvUserData.DataSource
        Dim dc As System.Data.DataColumn
        Dim dr As System.Data.DataRow
        Dim colIndex As Integer = 0
        Dim rowIndex As Integer = 0
```

```
        For Each dc In dt.Columns
            colIndex = colIndex + 1
            _excel.Cells(1, colIndex) = UCase(dc.ColumnName)
        Next
```

```
        For Each dr In dt.Rows
            rowIndex = rowIndex + 1
            colIndex = 0
            For Each dc In dt.Columns
                colIndex = colIndex + 1
                _excel.Cells(rowIndex + 1, colIndex) = dr(dc.ColumnName)
```

```
            Next
        Next
```

```

wSheet.Columns.AutoFit()
wSheet.Columns("P:P").ColumnWidth = 100
wSheet.Columns("P:P").RowHeight = 17

wSheet.Range("A1", "Z1").Font.Bold = True

Dim strMainPath As String =
    Configuration.ConfigurationManager.AppSettings("Path")
Dim strFilepath As String = strMainPath & "\\Data\\Student Master"
If IO.File.Exists(strFilepath) = False Then
    Directory.CreateDirectory(strFilepath)
End If

Try

    wBook.SaveAs(strFilepath & "\\" & Format(Date.Now, "dd-MMM-yyyy"))
    'lblExportPath.Text = strFilepath & "\\" & Format(Date.Now,
        "dd-MMM-yyyy")
Catch ex As Exception
    MessageBox.Show("Please close current excel sheet !!", strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
Exit Sub
End Try

wBook.Close()
_excel.Quit()
MessageBox.Show("Student Data is exported successfully !! at " &
    strFilepath & "\\" & Format(Date.Now, "dd-MMM-yyyy"), strTitle,
    MessageBoxButtons.OK, MessageBoxIcon.Information,
    MessageBoxDefaultButton.Button1)
Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
Finally
    Runtime.InteropServices.Marshal.ReleaseComObject(_excel)
    Me.Cursor = Cursors.Default
End Try
End Sub

End Class

```

Transaction Code:

```
Imports System.Text
Imports System.IO
```

```
Public Class frmTransection
```

```
    '---serial port to listen to incoming data---
```

```
    Private WithEvents serialPort As New IO.Ports.SerialPort
```

```
    '---tag ID read from the reader---
```

```
    Private tagID As String = String.Empty
```

```
    '---the time that the tag ID was recorded---
```

```
    Private timeRecorded As DateTime = Now
```

```
    '---COM port to listen to---
```

```
    Const COM As String = "COM5"
```

```
    '---filename of the log file---
```

```
    Const FILE_NAME As String = "D:\Attendance.csv"
```

```
    '---the interval before the employee record is cleared
```

```
    ' from the screen (in seconds)---
```

```
    Const INTERVAL As Integer = 3
```

```
    Dim email As String
```

```
    Dim ObjDataLayer As New DataAccessLayer
```

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

```
        If serialPort.IsOpen() Then
```

```
            serialPort.Close()
```

```
            serialPort.Dispose()
```

```
        End If
```

```
    End Sub
```

```
    Private Sub frmTransectionEntry_Activated(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles Me.Activated
```

```
        MDIForm.Panel1.SendToBack()
```

```
        MDIForm.Panel1.Dock = DockStyle.Fill
```

```
    End Sub
```

```
    Private Sub frmTransectionEntry_FormClosing(ByVal sender As Object, ByVal e
        As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
```

```
        MDIForm.Panel1.BringToFront()
```

```
        MDIForm.Panel1.Dock = DockStyle.Fill
```

```
    End Sub
```

```

Private Sub frmCreateUser_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    Try

        Me.ControlBox = False

        Timer1.Interval = INTERVAL * 1000 'convert to milliseconds
        '---open the serial port connecting to the reader---
        If serialPort.IsOpen Then
            serialPort.Close()
        End If
        Try
            With serialPort
                .PortName = COM
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
            End With
            serialPort.Open()
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
        fillAutoID()
        FillGridView()
        ClearAll()
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try

End Sub

Private Sub DataReceived(ByVal sender As Object, ByVal e As
    System.IO.Ports.SerialDataReceivedEventArgs) Handles
    serialPort.DataReceived
    '---when incoming data is received, update the TagID
    ' textbox---
    txtRFID.BeginInvoke(New myDelegate(AddressOf updateTextBox), New
        Object() {})
End Sub
'---update the Tag ID textbox---
Public Delegate Sub myDelegate()

```



```

Public Sub updateTextBox()
    '---for receiving plain ASCII text---
    If txtRFID.TextLength = 12 Then
        'txttagid.Text = ""
        With txtRFID
            .AppendText(serialPort.ReadExisting)
            .ScrollToCaret()
        End With
    Else
        With txtRFID
            .AppendText(serialPort.ReadExisting)
            .ScrollToCaret()
        End With
    End If

End Sub

Private Sub FillGridView()
    Try

        Dim dtDetails As New DataTable
        Dim strSQL As String = "select te.AutoID,te.TransectionID
            [Transection ID],te.CameraID [Camera ID],te.RFID,se.StudentID
            [Student ID],se.StudentName [Student Name],se.Class,se.CollegeName
            [College Name] ,convert(varchar(50),te.EntryDate,103) [Entry Date]
            from TransectionEntry te left join mstStudentEntry se on
            te.RFID=se.RFID "
        dtDetails = ObjDataLayer.FillDataTable(strSQL)
        DgvUserData.DataSource = dtDetails
        DgvUserData.Columns("AutoID").Visible = False
        lblTotal.Text = "Total Count = " & dtDetails.Rows.Count
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try
End Sub

Private Sub ClearAll()
    Try

        txtCameraID.Text = "Camera1"
        txtRFID.Text = ""

        fillAutoID()
    Catch ex As Exception

```

```

        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try

End Sub

Private Sub fillAutoID()
    Try
        Dim dsDomestic As New DataTable
        dsDomestic = ObjDataLayer.FillDataTable("select
            Max(cast(substring(TransectionID,12,50000) as int)) as
            TransectionID from TransectionEntry")

        If IsDBNull(dsDomestic.Rows(0)(0)) = True Then
            txtTransectionID.Text = "TRNID/Trn/-1"
        Else
            txtTransectionID.Text = "TRNID/Trn/-" & dsDomestic.Rows(0)(0) + 1
        End If

        Catch ex As Exception
            MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
                MessageBoxButtons.OK, MessageBoxIcon.Information,
                MessageBoxDefaultButton.Button1)
        End Try
    End Sub

Private Function Validation() As Boolean
    Validation = True
    Try
        Dim strMsg As New StringBuilder

        If txtCameraID.Text = "" Then
            strMsg.AppendLine("Please enter Transection Name")
            Validation = False
        End If

        If txtRFID.Text = "" Then
            strMsg.AppendLine("Please enter RFID")
            Validation = False
        End If

        If txtRFID.Text.Length <> 12 Then

```

```

        strMsg.AppendLine("Please enter Valid RFID")
        Validation = False
    End If

    If strMsg.Length > 0 Then
        MessageBox.Show(strMsg.ToString, strTitle, MessageBoxButtons.OK,
            MessageBoxIcon.Information, MessageBoxDefaultButton.Button1)
        Return False
    End If
    Return Validation
Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Information,
        MessageBoxDefaultButton.Button1)
End Try
End Function

Private Sub txtRFID_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles txtRFID.TextChanged
    If txtRFID.Text.Length = 12 Then
        Dim strSQL As String
        Dim dtLogin As New DataTable
        Dim dtRFID As New DataTable

        strSQL = "select * from mstStudentEntry where RFID='" +
            txtRFID.Text.Trim() + "'"
        dtRFID = ObjDataLayer.FillDataTable(strSQL)
        If dtRFID.Rows.Count > 0 Then
            strSQL = "Insert INTO
                TransectionEntry(TransectionID,CameraID,RFID ,EntryDate) " & _
" Values('" + Trim(txtTransectionID.Text).Replace("'", "''") + "', '"
            + Trim(txtCameraID.Text).Replace("'", "''") + "', '" & _
            Trim(txtRFID.Text.Trim()).Replace("'", "''") & "', '" & Date.Now &
            "'"")

            ObjDataLayer.ExecuteNonQuery(strSQL)
            strSQL = "select * from TransectionEntry where RFID='" +
                txtRFID.Text.Trim() + "'"
            dtLogin = ObjDataLayer.FillDataTable(strSQL)
            If dtLogin.Rows.Count > 1 Then
                strSQL = "delete from TransectionEntry where RFID='" +
                    txtRFID.Text.Trim() + "'"
            End If
        End If
    End If
End Sub

```

```

        ObjDataLayer.ExecuteNonQuery(strSQL)
    End If
    frmCreateUser_Load(sender, e)
Else
    MessageBox.Show("Invalid RFID ", strTitle, MessageBoxButtons.OK,
        MessageBoxIcon.Information, MessageBoxDefaultButton.Button1)
    txtRFID.Text = ""
End If

End If
End Sub
End Class
Data access code:
Imports System.Data
Imports System.Data.SqlClient

Imports System.Text.RegularExpressions

Public Class DataAccessLayer

    Public strSQLPath As String =
        System.Configuration.ConfigurationSettings.AppSettings("ServerPath")
    Public strConn As SqlConnection = New SqlConnection(strSQLPath)

    Public Function ValidateEmailId(ByVal emailTextBox As TextBox) As Boolean
    Try
        Dim emailAddress As String = Trim(emailTextBox.Text)
        If emailAddress <> "" Then
            Dim pattern As String =
                "^[a-zA-Z] [\w\.-]*[a-zA-Z0-9]@[a-zA-Z0-9] [\w\.-]*[a-zA-Z0-9] \.[a-zA-Z] [a-
            Dim emailAddressMatch As Match = Regex.Match(emailAddress,
                pattern)
            If emailAddressMatch.Success Then
                ValidateEmailId = True
            Else
                ValidateEmailId = False
                emailTextBox.Focus()
                emailTextBox.BackColor = Color.MediumAquamarine
            End If
        End If
    Catch
    End Try
End Function
End Class

```

```

        MsgBox("This Email Address is not Valid..!",
            MsgBoxStyle.Information)
        emailTxtBox.BackColor = Color.White
        Exit Function
    End If
Else
    Return True
End If
Return True
Catch ex As Exception
    MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
        MessageBoxButtons.OK, MessageBoxIcon.Error,
        MessageBoxDefaultButton.Button1)
End Try
End Function

Public Sub ExecuteNonQuery(ByVal strSQL As String)
    Try
        Dim cmd As New SqlCommand(strSQL, strConn)

        If strConn.State = ConnectionState.Closed Then
            strConn.Open()
        End If
        cmd.ExecuteNonQuery()
        If strConn.State = ConnectionState.Open Then
            strConn.Close()
        End If
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try
End Sub

Public Function FillDataTable(ByVal strSQL As String) As DataTable
    FillDataTable = Nothing
    Try
        Dim dtTable As New DataTable
        Dim dtAdp As New SqlDataAdapter(strSQL, strConn)
        dtAdp.Fill(dtTable)
        Return dtTable
    Catch ex As Exception
        MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
            MessageBoxButtons.OK, MessageBoxIcon.Information,
            MessageBoxDefaultButton.Button1)
    End Try
End Function

```

```
End Try
End Function
```

```
Public Sub FillNewCombo(ByVal cmb As ComboBox, ByVal strSQL As String, ByVal
    strBool As Boolean)
    'Dim SqlStr As String = "Select * From EqpModelMaster where
        EquipmentID='" + SelEqpId + "' And CostINR IS NOT Null;"
    cmb.Items.Clear()

    Dim cmd As New SqlCommand(strSQL, strConn)
    strConn.Open()
    Dim dst As New SqlDataAdapter(strSQL, strConn)
    Dim dr As SqlDataReader
    Try
        dr = cmd.ExecuteReader
        While dr.Read
            cmb.Items.Add(dr(0))
        End While
        strConn.Close()
        If strBool = True Then
            cmb.Items.Add("ALL")
            'cmb.Text = "All"
        End If

        Catch ex As Exception
            MsgBox(ex.Message)
            strConn.Close()
        End Try
    End Sub
```

```
Public Sub FillCombo(ByRef cmb As ComboBox, ByVal strSQL As String, ByVal
    DispMember As String, ByVal ValMember As String)
    Try
        Dim ds As New DataTable
        ds = FillDataTable(strSQL)

        cmb.DataSource = ds

        cmb.DisplayMember = DispMember
        cmb.ValueMember = ValMember

        Catch ex As Exception
            MessageBox.Show(ex.Message & ex.Source & ex.StackTrace, strTitle,
                MessageBoxButtons.OK, MessageBoxIcon.Error,
```

```

        MessageBoxDefaultButton.Button1)
    Finally
        If strConn.State = ConnectionState.Open Then
            strConn.Close()
        End If
    End Try
End Sub

End Class

Camera access code:
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class CameraView : System.Web.UI.Page
{
    DataAccessLayer DataLayer = new DataAccessLayer();

    protected void Page_Load(object sender, EventArgs e)
    {
        txtStudentID.Text = Session["StudentID"].ToString();
        lblMessage.Text = "";
        iframe1.Attributes["src"] = "http://192.168.1.5:8080/browser.html";
        if (txtStudentID.Text != "")
        {
            string strCamera ;
            string strSQL;

            DataTable dtTable = new DataTable();
            strSQL = "select te.AutoID,te.TransectionID [Transection
                ID],te.CameraID [Camera ID],te.RFID,se.StudentID [Student
                ID],se.StudentName [Student Name],se.Class,se.CollegeName [College
                Name] ,convert(varchar(50),te.EntryDate,103) [Entry Date] from
                TransectionEntry te left join mstStudentEntry se on te.RFID=se.RFID
                where se.StudentID='" + txtStudentID.Text.Trim() + "' ";

```

```

dtTable = DataLayer.FillDataTable(strSQL);

if (dtTable.Rows.Count > 0)
{
    strCamera = dtTable.Rows[0]["Camera ID"].ToString();
    if (strCamera == "Camera1")
    {
        iframe1.Attributes["src"] =
            "http://172.16.41.88:8080/browser.html";
        lblMessage.Text = "Camera 1";
    }
    else
    {
        iframe1.Attributes["src"] =
            "http://192.168.1.8:8080/browser.html";
        lblMessage.Text = "Camera 2";
    }
}
else
{
}

}
}

```


4.2 Test Cases

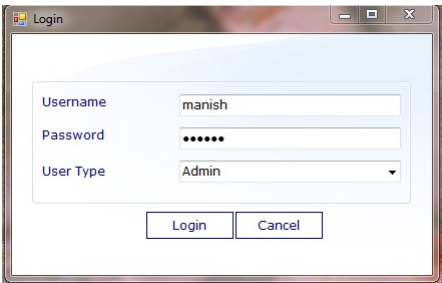
Test Case	Check Item	Test case Description	Steps to Execute	Test Data / Input	Expected Result
TC-001	Log-in Page	Leave all fields as blank and click Log-in button	Click Log-in		By leaving all fields as blank and on click Log-in button then mandatory text "Invalid Login" should appear in front of User
TC-002	Uusername	Enter Invalid Uusername	Enter incorrect Uusername	Uusername : ABC	By entering invalid Uusername then an error message should appear as " Please Enter Valid Uusername"
TC-003	Uusername	Enter valid Uusername	Enter correct Uusername	Uusername : Rahul	It should allow the user to proceed
TC-004	Password	Enter Text	Enter text on the given field	Password :1234	The password field should display the encrypted format of the text typed as (****)
TC-005	Password	Enter wrong password	Enter invalid Password	Password : ***	By entering invalid password then an error message should appear as " Incorrect Password "
TC-006	Password	Enter Correct password	Enter valid Password	Password : *****	It should allow the user to proceed
TC-007	Log-in button	Correct Inputs	Click Log-in		It should lead the user to the respective page
TC-008	Registration	Load correct page	Click Registration		On click "Registration" page should redirect to the User Registration page.

4.3 Result Metrics

Table 4.1: Result Metrics

Video quality	Bandwidth	Time
480x320	128kbps	3.03sec
640x480	240kbps	2.08sec
720x480	600kbps	1.02sec
768x432	800kbps	1.01sec
HD	1mbps	0.5sec

4.4 Result snapshots



A login window titled "Login" with a light blue background. It contains three input fields: "Username" with the text "manish", "Password" with masked characters "*****", and "User Type" with a dropdown menu showing "Admin". Below the fields are two buttons: "Login" and "Cancel".

Figure 4.1: Teacher Login Window

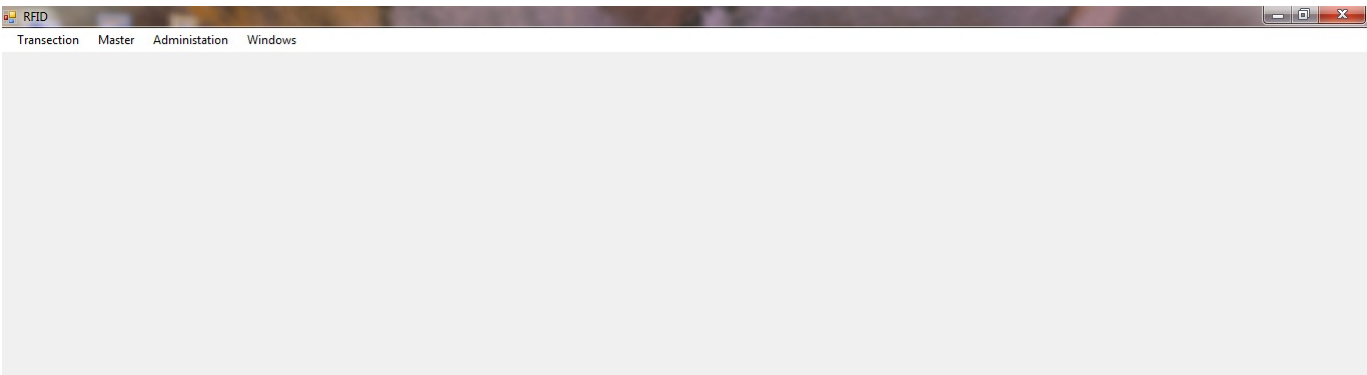
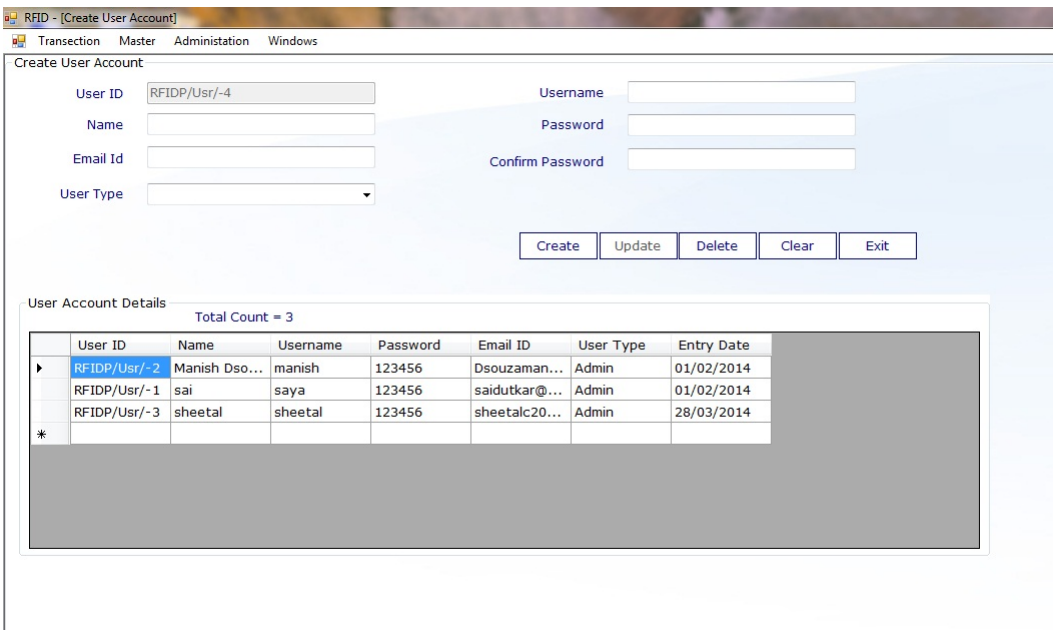


Figure 4.2: User Workspace



A "Create User Account" form within the "RFID" application. The form has a title bar "RFID - [Create User Account]" and a menu bar "Transaction Master Administration Windows". It contains several input fields: "User ID" (pre-filled with "RFIDP/Usr/-4"), "Name", "Email Id", "User Type" (dropdown), "Username", "Password", and "Confirm Password". Below these fields are buttons for "Create", "Update", "Delete", "Clear", and "Exit". At the bottom, there is a "User Account Details" section with a "Total Count = 3" and a table listing existing users.

	User ID	Name	Username	Password	Email ID	User Type	Entry Date
▶	RFIDP/Usr/-2	Manish Dso...	manish	123456	Dsouzaman...	Admin	01/02/2014
	RFIDP/Usr/-1	sai	saya	123456	saidutkar@...	Admin	01/02/2014
	RFIDP/Usr/-3	sheetal	sheetal	123456	sheetalc20...	Admin	28/03/2014
*							

Figure 4.3: Create User Form

RFID - [Student Entry]

Transaction Master Administration Windows

Student Entry

Student ID: RFIDS/Stu/-6

Student Name:

RFID:

Email ID:

Parent Username:

Parent Password:

Phone:

Class:

College Name:

Address:

Create Update Delete Clear Exit

Student Details

Total Count = 3

	Student ID	Student N...	RFID	EmailID1	Phone1	Class	College Na...	Address	ParentUse...	ParentPas...	Entry
▶	RFIDS/Stu/-5	Apoorva	FE0034DE2...	schaudhari...	987654321	BE	SPIT	mumbai	Sheetal	123456	28/03
	RFIDS/Stu/-4	Manish Dso...	8400810EF...	dsouzamani...	8390895729	BEIT	SPIT	VIRAR	manish	manish	27/03
	RFIDS/Stu/-3	Saya	FE006EC9D...	Saya@gmai...	86632325	BE	SPIT	Vasai	sai	sai	10/03
*											

Figure 4.4: Student Entry Form

RFID - [Attendance]

Transaction Master Administration Windows

Transaction Entry

Attendance ID: RFIDS/Atn/-5

Student Name: Apoorva

In Time:

Date:

Out Time:

In Time Out Time Exit

	Attendanc...	Student N...	In Time	Date	Out Time
▶	RFIDA/Atn/-1	Saya	11:28 AM	28/03/2014	14:30 PM
	RFIDS/Atn/-2	Manish Dso...	11:53 AM	28/03/2014	11:53 AM
	RFIDS/Atn/-3	Manish Dso...	14:30 PM	28/03/2014	14:31 PM
	RFIDS/Atn/-4	Apoorva	14:59 PM	28/03/2014	14:59 PM
*					

Figure 4.5: Attendance Updation Form

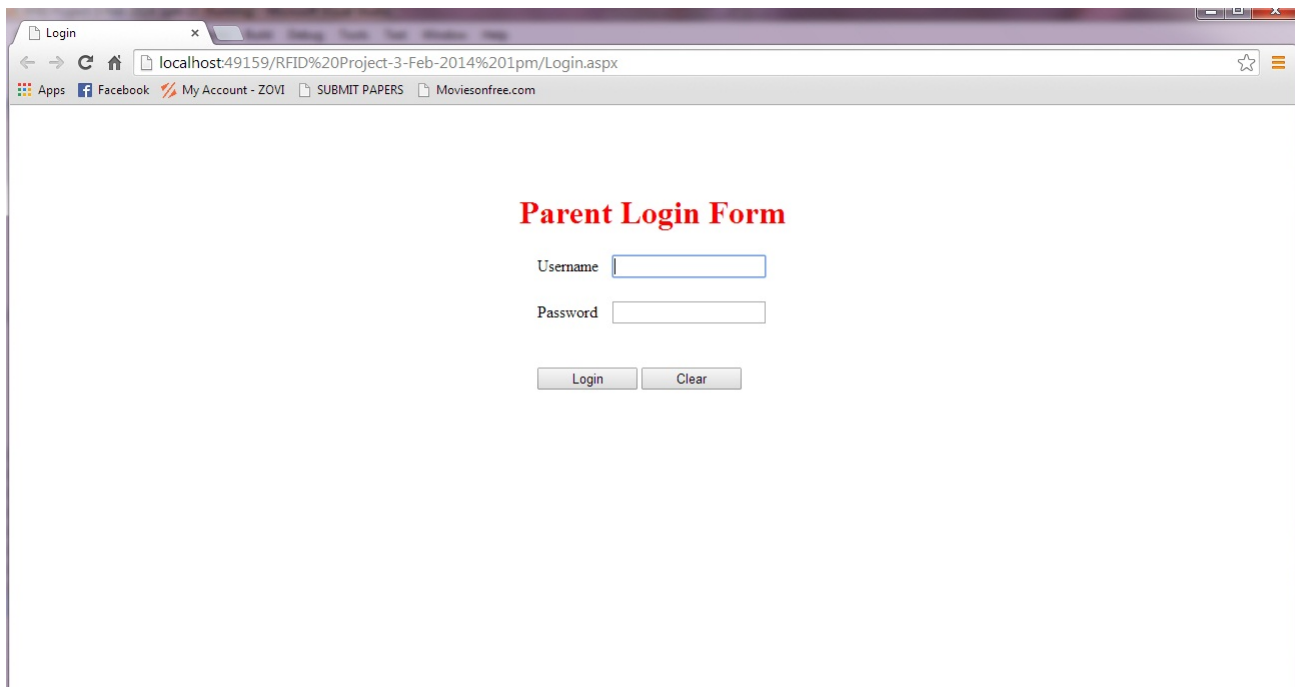


Figure 4.6: Parent Login Page

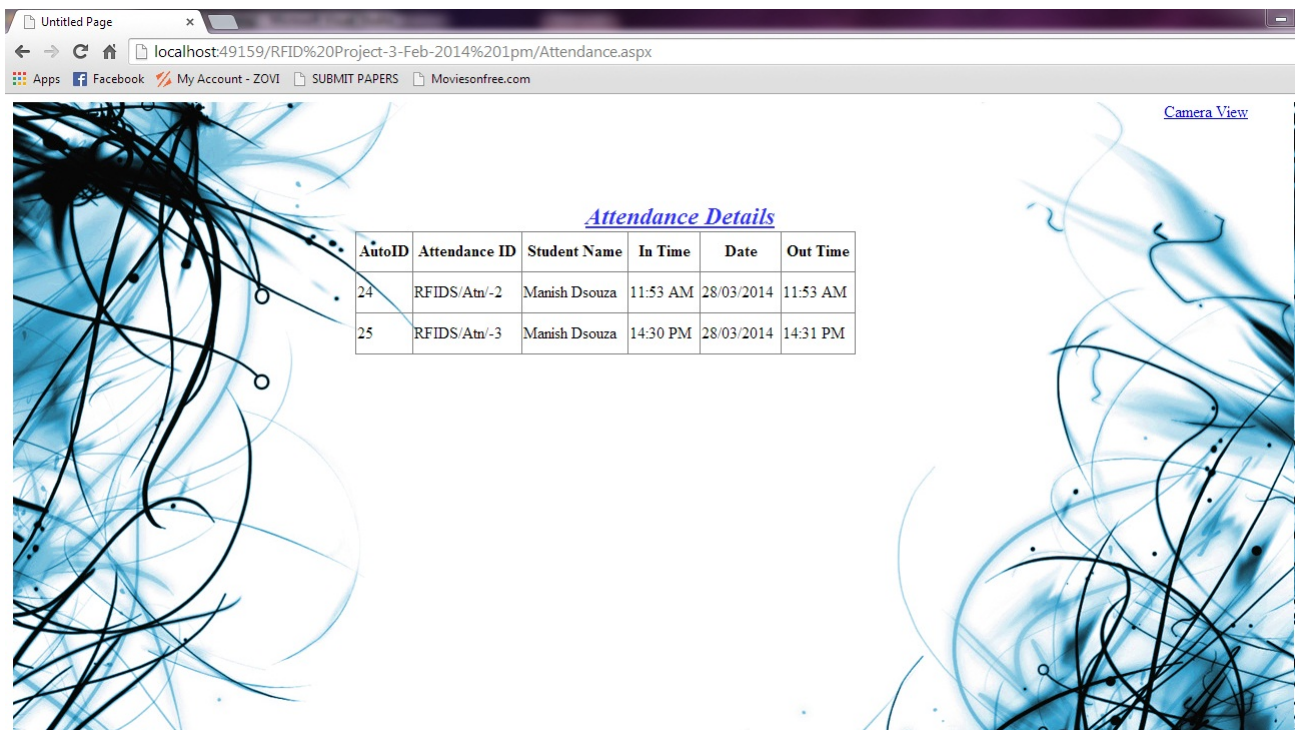


Figure 4.7: Attendance Report

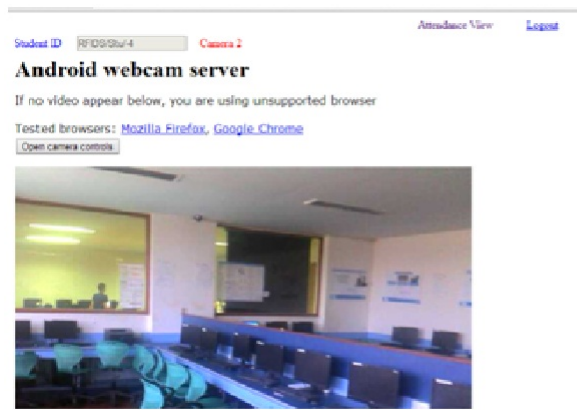


Figure 4.8: Live Camera Feed

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This software project development work was educating to us. Lot of things we have learned during this period. Coordination, correlation is most important while designing database, tables, menu designs etc. At some stages we found bottle necks when our team leader, Project Guide and at last the world wide web was at our service. Phases or stages of System analysis and design practically understood while designing the system. Understanding user requirements has most importance. We also learned naming conventions for Fields, tables, forms, controls and objects.

5.2 Future Scope

- In future we would like add more features and make more user- friendly
- We would like to make more accurate and precise and have a high end input system.
- It should be merged with more complex systems.
- It should also be more enhanced regarding the security issues.

5.3 Adding Motion Detection And Alarm Generation

There are many approaches for motion detection in a continuous video stream. All of them are based on comparing of the current video frame with one from the previous frames or with something that we'll call background.

Algorithm

The most efficient algorithms are based on building the so called background of the scene and comparing each current frame with the background. There are many approaches to build the scene, but most of them are too complex. Let's assume that we have an original 24 bpp RGB image called current frame (image), a grayscale copy of it (currentFrame) and a background frame also gray scaled (backgroundFrame). At the beginning, we get the first frame of the video sequence as the background frame. And then we'll always compare the current frame with the background one. But it will give us the results, which we obviously don't want very much. Our approach is to "move" the background frame to the current frame on the specified amount (I've used 1 level per frame). We move the background frame slightly in the direction of the current frame - we are changing colors of pixels in the background frame by one level per frame.

```
// create filter
MoveTowards moveTowardsFilter = new MoveTowards( );
// move background towards current frame
moveTowardsFilter.OverlayImage = currentFrame;
Bitmap tmp = moveTowardsFilter.Apply( backgroundFrame );
// dispose old background
backgroundFrame.Dispose( );
backgroundFrame = tmp;
```

And now, we can use the same approach we've used above. But, let me extend it slightly to get a more interesting result.

```
// create processing filters sequence
FiltersSequence processingFilter = new FiltersSequence( );
processingFilter.Add( new Difference( backgroundFrame ) );
processingFilter.Add( new Threshold( 15 ) );
processingFilter.Add( new Opening( ) );
processingFilter.Add( new Edges( ) );
// apply the filter
Bitmap tmp1 = processingFilter.Apply( currentFrame );

// extract red channel from the original image
IFilter extrachChannel = new ExtractChannel( RGB.R );
Bitmap redChannel = extrachChannel.Apply( image );
// merge red channel with moving object borders
Merge mergeFilter = new Merge( );
mergeFilter.OverlayImage = tmp1;
Bitmap tmp2 = mergeFilter.Apply( redChannel );
// replace red channel in the original image
ReplaceChannel replaceChannel = new ReplaceChannel( RGB.R );
replaceChannel.ChannelImage = tmp2;
```



```
Bitmap tmp3 = replaceChannel.Apply( image );
```



Figure 5.1: Motion Detection And Alarm Generation

Motion Alarm

It is pretty easy to add motion alarm feature to all these motion detection algorithms. Each algorithm calculates a binary image containing difference between current frame and the background one. So, the only we need is to just calculate the amount of white pixels on this difference image.

```
// Calculate white pixels
private int CalculateWhitePixels( Bitmap image )
{
    int count = 0;
    // lock difference image
    BitmapData data = image.LockBits( new Rectangle( 0, 0, width, height ),
        ImageLockMode.ReadOnly, PixelFormat.Format8bppIndexed );
    int offset = data.Stride - width;
    unsafe
    {
        byte * ptr = (byte *) data.Scan0.ToPointer( );
        for ( int y = 0; y < height; y++ )
        {
            for ( int x = 0; x < width; x++, ptr++ )
            {
                count += ( (*ptr) >> 7 );
            }
            ptr += offset;
        }
    }
    // unlock image
    image.UnlockBits( data );
}
```



```

    return count;
}

```

For some algorithms it could be done even simpler. For example, in blob counting approach we can accumulate not the white pixels count, but the area of each detected object. Then, if the computed amount of changes is greater than a predefined value, we can fire an alarm event.

Video Saving

There are many different ways to process motion alarm event: just draw a blinking rectangle around the video, or play sound to attract attention. But, of course, the most useful one is video saving on motion detection. In the demo application I was using the AVIWriter class, which uses Video for Windows interop to provide AVI files saving capabilities. Here is the small sample of using the class to write small AVI file, which draw diagonal line:

```

SaveFileDialog sfd = new SaveFileDialog( );
if ( sfd.ShowDialog( ) == DialogResult.OK )
{
    AVIWriter writer = new AVIWriter( "wmv3" );
    try
    {
        writer.Open( sfd.FileName, 320, 240 );
        Bitmap bmp = new Bitmap( 320, 240, PixelFormat.Format24bppRgb );
        for ( int i = 0; i < 100; i++ )
        {
            bmp.SetPixel( i, i, Color.FromArgb( i, 0, 255 - i ) );
            writer.AddFrame( bmp );
        }
        bmp.Dispose( );
    }
    catch ( ApplicationException ex )
    {
    }
    writer.Dispose( );
}

```

Bibliography

- [1] Michael, K., Roussos, G., Huang, G. Q., Gadh, R., Chattopadhyay, A., Prabhu, S., & Chu, , " Planetary-scale RFID services in an age of uberveillance", *Proceedings of the IEEE*, vol. 98, 2010,pp. 1663-1671.
- [2] Bigger Monster, Weaker Chains,"The Growth of an American Surveillance Society", *American Civil Liberties Union. January , January 15*, vol. 3, No.1,pp. 75-85.
- [3] Park, J. and Kim, "FBI's Secret Spyware Tracks Down Teen Who Made Bomb Threats", *Wired Magazine, April 2010*, , July 18, 2007
- [4] CALEA: Frequently Asked Questions, *Electronic Frontier Foundation (website)*. Retrieved March 14, 2009.
- [5] Ambinder, Marc , " Pinwale And The New NSA Revelations," *The Atlantic*. Retrieved June 30, 2009
- [6] Kuhn, M.G. (2004), "Electromagnetic Eavesdropping Risks of Flat-Panel Displays.," *4th Workshop on Privacy Enhancing Technologies*,pp. 2325.
- [7] Risen, James; Lichtblau, Eric (June 16, 2009), "E-Mail Surveillance Renews Concerns in Congress," *New York Times*. pp. A1. Retrieved June 30, 2009.