****

**UNIVERSITY OF MUMBAI**

**A Project Report on**

**Timezone-Spy**

Submitted by

Sucheta Naik : [seat no.]

March - 2013

Under the guidance of

Prof. Sheetal Asodekar

Submitted in partial fulfillment of the requirements for qualifying

Examination (B.Sc. - I.T., Semester - VI (Rev) Examination)

NIRMALA MEMORIAL COLLEGE OF COMMERCE AND SCIENCE

DS Road, Asha Nagar, Thakur Complex, Kandivali (East), Mumbai, 400101.

**PREFACE:**

This book is meant to serve as a guide for our project which is a computer application. This book comprehensively covers all major aspects of our project. We are glad to present our project through this report in which details of development & user manual to use this software are included. It also presents the database used, methodology adopted and the flow of the data throughout the project.

It describes the various feasibilities cost & technical requirement for the software. It desks the actual input output views (screenshots) which the user would use to interact with the software system diagram & data dictionary are provided for better understanding of the project.

**ACKNOWLEDGEMENT**

I express my thanks and gratitude to Almighty God, my parents and other family members and friends without whose uncontained support, I could not have made this career in Timezone-spy application.

I wish to place on my record my deep sense of gratitude to my project guide, Prof. Sheetal Asodekar for her constant motivation and valuable help through the project work. Express my gratitude to Prof. Sumathi Rajkumar, for her valuable suggestions and advices throughout the project. I also extend my thanks to other faculties for their cooperation during my project.

Finally I would like to thank my friends for their cooperation to complete this project.

**SYNOPSIS OF THE PROJECT**

**INTRODUCTION:**

This project is aimed at developing a tracking application useful to be used in an organization. The Timezone-Spy is a computer application that can be accessed throughout the organization. This application can be used for tracking computers connected to the network. There are features like set break, get break, etc. for the users using this application.

There are some main components of this project. They are given below:

**ADMINISTRATOR:**

Administrator can perform these following tasks:

1. **User Maintenance:**

Users of this application can be employee, student, etc. Employees can be recruited by the administrator. Administrator is authenticated to add user and track user.

1. **Role Management:**

In this application role management can be done by the administrator only. The roles played by the administrator can be as follows:

* The administrator can send messages to all the users that are currently logged in the application.
* The administrator can track the user by setting a time for taking screenshot automatically by the application.

1. **Component Maintenance**

Here component in the sense of software which are using in this application. The component can be classified into different types and modules

1. Operating System Software
2. Programming Software
3. Networking Software
4. Data Base Software

**OPERATIONAL ADMINISTRATOR:**

In this application named as Timezone-Spy, the one who is responsible for solving any defect in the application is called as **Operational Administrator**. Every operational administrator will come to know is any user has logged-in or logged-out. The messages received by all the users will be sent by this operational administrator. After receiving screenshots from the user’s desktop, this administrator can see it on desktop.

**OPERATOR:**

In this application named as Timezone-Spy, the operator can be employee, student, etc. An operator’s role is to set break i.e. to click on set break button so that the administrator can be notified that the user is on break, end break i.e. to click on end break button so that the administrator can be notified that the user has ended break and can enter break reason in order to inform the operator about the reason for the break taken.

**MERITS OF THE APPLICATION:**

The project is identified by the merits of the system offered to the user. The merits of this project are as follows:

* This application is a windows form application.
* This project offers the admin to track the usage of all the computers connected to the server by just one click.
* The admin is more concerned about the usage of the computers used by the users.
* Sometimes the admin finds in the later stages of using project that the message needs to be updated and the new message can be updated but the previous message sent to all the users through this application.
* Admin is provided the option of monitoring the records he entered earlier. He can see the desired records of the users.
* Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
* Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.
* Easier and faster data transfer through latest technology associated with the computer and communication.
* Through these features it will increase the efficiency, accuracy and transparency

**LIMITATIONS:**

* The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
* The reports generated are stored in computers and if the hard disk is crash or if the data is lost the reports generated will also be lost.
* The user added by the admin cannot be deleted as it is required to keep the users track as database.
* The message sent by the admin to the users cannot be removed.

**PROJECT SCHEDULE AND NETWORK DIAGRAM:**

1. **WORK BREAKDOWN STRUCTURE:**

The table below shows the expenditure profile among the activities in the Work Breakdown Structure:

|  |  |
| --- | --- |
| **ACTIVITY** | **BUDGET (in %)** |
| Requirements analysis | 5 |
| Product design | 13 |
| Programming | 45 |
| Test planning | 8 |
| Verification and validation | 15 |
| Configuration management and quality assurance | 6 |
| Manual | 8 |

**b. GANTT CHART**

* A Gantt chart is a type of bar chart that illustrates a project schedule.
* Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project.
* A Gantt chart is a graphical representation of the duration of tasks against the progression of time.
* It is a useful tool for planning and scheduling projects

.

* A Gantt chart is helpful when monitoring a project's progress

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | TASK NAME  (weeks) | November  2012  4th | December  2012  1 2 3 4 | January  2013  1 2 3 4 | February  2013  1 2 3 4 | March  2013  1 2 3 4 |
| 1. | Project Planning |  |  |  |  |  |
| 2. | System Analysis |
| 3. | System Design |
| 4. | Coding |
| 5. | Validation Testing |
| 7. | Documentation and  User Manual |

Planned

|  |  |  |
| --- | --- | --- |
| Sr. No | Contents | Page Number |
| 1. | ANALYSIS PHASE   * 1. Requirement Gathering      1. Problem Statement      2. Objective And Scope Of Projects      3. Problems With Existing System \*      4. Advantage of Proposed System   2. Feasibility Study      1. Operational Feasibility      2. Technical Feasibility      3. Economical Feasibility      4. Behavior Feasibility      5. Cost Benefit Analysis      6. Gantt Chart      7. PERT Chart   3. Requirement Specification      1. Functional Requirement      2. Technical Requirement      3. Software Requirement Specification      4. Requirement decomposition Diagram   4. Tools and Technology |  |
| 2 | DESIGN PHASE  2.1 Detailed Lifecycle Of Project(Logical Design)  2.1.1 E-R Diagram  2.1.2 Context Diagram  2.1.3 Event Table  2.1.4 DFD  2.1.4.1 DFD Level 0, Level 1, and so on  2.2 Physical Design  2.2.1 System Flow Chart |  |
| 3 | IMPLEMENTATION PHASE  3.1 Coding/Building  3.1.1 Screen layouts  3.1.2 Deployment  3.2 Testing Phase  3.2.1 Testing And Its Types  3.2.2 Methodologies Adopted For Testing |  |
| 4. | MAINTAINANCE AND EVALUATION  4.1 System Maintenance And Future Enhancement  4.2 User manual |  |
| 5 | ANNEXURE  5.1 Data Dictionary  5.2 List of Abbreviations, figures, tables  5.3 References: bibliography, websites used. |  |

**INTRODUCTION:**

This project is aimed at developing a computer application useful for offices, colleges, cyber etc. The project named as Timezone-Spy is windows based application that can be accessed throughout the server using lan i.e. local area network. This application can be used for enhancing the interaction between the users and the admin. There are features like sending messages, tracking users taking screenshots, report generators etc. in this application.

1. **ANALYSIS PHASE**
   1. **REQUIREMENT GATHERING :**

**1.1.1 PROBLEM STATEMENT:**

This project is aimed at developing a computer application useful for offices, colleges, cyber etc. The project named as Timezone-Spy is windows based application that can be accessed throughout the server using lan i.e. local area network. This application can be used for enhancing the interaction between the users using the computers and the admin. There are features like sending messages, taking screenshots, report generators in this application.

The development of this application contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

User friendliness is provided in the application with various controls. The system makes the overall project much easier and flexible to the end user. There is no risk of data mismanagement at any level while the project development is under process. It provides high level of security. The system interface provides good environment to solve any kind of error.

* + 1. **OBJECTIVE AND SCOPE OF PROJECTS:**

**Objective:**

The objective is to enhance the security and decrease the communication gap between the administrator and the user with a common lan i.e. local area network or with a common server.

**Scope of Projects:**

User record can be checked in future it can be used as users working record screenshots can be viewed later as it is better and improved way to have record if user. It automatically takes screenshots. The data which is in the format of image is in kept in the form of binary data.

* + 1. **PROBLEMS WITH EXISTING SYSTEM:**

Timezone-Spy is a windows-based application designed to help an admin keep track of user’s task via local area network. The system was designed specifically keeping the offices and colleges in mind as it enhances the security by allowing the admin to track the users connected to the network.

It provides one roof solution for some security issues in the software development. It is a helpful system to trace users within the network and keep an eye over their acitivities via installed tool. It acts as a means to transmit the logged-in time logged-out time of the user with accuracy. It is a tool that helps in providing messages to the entire user by the admin in just one click. This tool also sends auto screenshots to administrator via hosted data.

In addition to all these, it provides a user friendly interface on both client and the administrator side. Its database is created very day so it is gives load to the operating system containing those databases. Also, as this application has been created to look over organizational structure, it does not involve user delete option.

* + 1. **ADVANTAGE OF PROPOSED SYSTEM**

The project is identified by the merits of the system offered to the user. The advantages of this project are as follows: -

* It’s a windows-enabled project.
* This project offers users to enter the data through simple and interactive forms. This helps the users to enter the desired information through so much simplicity.
* The admin creates id and password for the user which enhances the security by not allowing unauthorized user to use the application.
* The admin can update messages.
* The application has a calendar through which the user’s task can be shown according to the date.
* This project is bound to be simple and very friendly as per the user is concerned. That is, we can say that the project is user friendly which is one of the primary concerns of any good project.
* Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
* Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.
* Easier and faster data transfer through latest technology associated with the computer and communication.

Through these features it will increase the efficiency, accuracy and transparency.

**1.2** **FEASIBILITY STUDY:**

The project feasibility is concerned with expected benefits. An important outcome of the preliminary investigation is the destination that the proposed system is feasible.

There are three aspects in the feasibility study portion of the preliminary investigation**:**

1. **Operational Feasibility.**
2. **Technical Feasibility.**
3. **Economical Feasibility.**
4. **Behavioral Feasibility.**

**1.2.1 OPERATIONAL FEASIBILITY:**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**1.2.2 TECHNICAL FEASIBILITY:**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

The current system developed is technically feasible. It is windows based user interface. Thus it provides an easy access to the users. The user can be added by the admin only. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hardware requirements for the development of this project are not many and are already available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

**1.2.3 ECONOMICAL FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available. There is nominal expenditure and economical feasibility for certain.

1. **Effort Applied (E)** = a(KLOC)b[man-months]
2. **Development Time (D)** = c(Effort Applied)d[month]
3. **People required (P)** = Effort Applied / Development Time[count]
4. **Total Line of Code=** Approx. 3000(LOC)

Approx. 3(KLOC)

1. **Total Number of Hours =** Approx. 360 Hr. (3 ½ Month)

**3 Total number of People =** 2

* + 1. **BEHAVIOR FEASIBILITY**

Behavioral feasibility is the measure that how effective the user uses the system. The behavioral efficiency is one of the major factors of feasibility analysis. The new proposed system is easy to operate, convenient in maintenance and effective in its working. System should be...

1. Easy to operate (user friendly).

2. Existing user are not affected in anyway because easy environment is provided

3. Performance of software is accurate and fast to response.

**1.2.5 COST BENEFIT ANALYSIS**

Cost Benefit Analysis allows finding the cost of the actual software application created. Using COCOMO Model we can find the Cost, which was created by W.Bohem. Here using Basic COCOMO Model cost analysis is done.

The basic COCOMO equations take the form:

1. **Effort Applied (E)** = a(KLOC)b[man-months]
2. **Development Time (D)** = c(Effort Applied)d[month]
3. **People required (P)** = Effort Applied / Development Time[count]

* **KLOC** is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a, b, c and d are given in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software project** | ***A*** | **b** | ***c*** | ***d*** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |
|  |  |  |  |  |

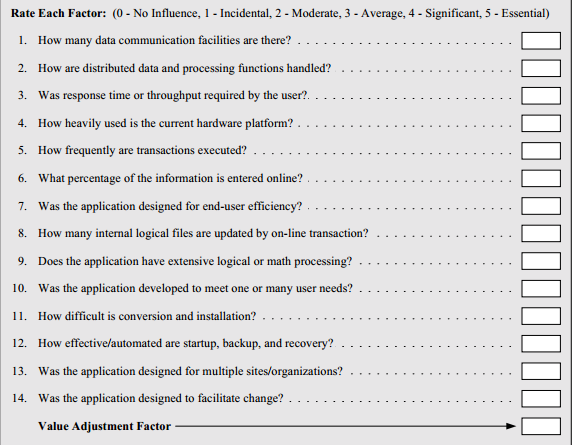
**Function Points Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Count | Simple | Average | Complex | Total |
| Number of user input | 40 | x 3 | 40x 4 | x 6 | 160 |
| Number of user output | 11 | x 4 | 11x 5 | x 7 | 55 |
| Number of Queries | 1 | x 3 | 1x 4 | x 7 | 4 |
| Number of files | 10 | 2 x 7 | 10x 10 | x15 | 100 |
| Number of Interface | 1 | x 5 | 1x 10 | x 15 | 10 |
| Total = | | | | | 329 |

Unadjusted Function Point Count = 329

Assuming the level of complexity and values of the software, we are calculating the adjusted Function Points.

**Calculating Adjusted Function Points:**



37

3

0

3

2

3

3

3

3

1

4

3

4

2

3

**Function Point Estimation:**

Function Points=Unadjusted Function point x (0.65 + 0.01x Value Adjustment Factor)

=329 x (0.65 + 0.01 x 37)

=335.58

i.e. 336 approximately.

**Approximate number of LOC for the C# language is:**

**Total Project Size**= 40\*336

= 13440 LOC.

=13.44 KLOC.

1. **Total Line of Code=** Approx. 13440(LOC)

Approx. 13.44(KLOC)

1. **Total Number of Hours =** Approx. 360 Hr (3 ½ Month )
2. **Total number of People =** 2

* So Effort Applied can be calculated as

**Effort Applied=** a (KLOC) b

= 2.4(13.44) 1.05

= 36.7308

**Effort Applied =** Approx. 36.7308

* + 1. **Pert Chart:**
* A PERT chart is a graphic representation of a project’s schedule, showing the sequence of tasks, which tasks can be performed simultaneously, and the critical path of tasks that must be completed on time in order for the project to meet its completion deadline.
* The chart can be constructed with a variety of attributes, such as earliest and latest start dates for each task, earliest and latest finish dates for each task, and slack time between tasks.
* A PERT chart can document an entire project or a key phase of a project.
* The chart allows a team to avoid unrealistic timetables and schedule expectations, to help identify and shorten tasks that are bottlenecks, and to focus attention on most critical tasks.

**1.3Requirement Specification:**

**1.3.1Functional Requirement:**

1. A user should be able to:

* Enter id in the application.
* Enter password in the application.
* Click on set break button while leaving the desk for break.
* Specify reason for break taken.
* Click on end break for resuming the work.
* The user should logout after he finishes work

2. The Application Admin should be able to do the following tasks

* Add a new user.
* Assign password for new user created.
* Track the user’s login time, logout time when needed.
* Send the user’s message connected to the server or to the local area network i.e. LAN.

The system after careful analysis has been identified to be presented with the following modules:

**The modules involved are:**

1. Administrator.

2. Operational Administrator.

3. Operator.

4. Search.

5. Reports.

6. Registration and Authentication.

**1. ADMINISTRATOR:**

Administrator can perform these following tasks

**User Maintenance**

Users of this system can be employee, student, etc. Employees can be recruited by the administrator only.

**Component Maintenance**

Here component in the sense of software which are using in this organization. The component can be classified into different types and modules

* Operating System Software
* Programming Software
* Networking Software
* Data Base Software

The above all tasks should be performed by Admin only.

**2. OPERATIONAL ADMINISTRATOR:**

In this application named as Timezone-Spy, the one who is responsible for solving any defect in the application is called as Operational administrator. Every operational administrator has a unique login id and password. The messages received by all the users will be sent by this operational administrator. After receiving screenshots from the user’s desktop, this administrator can affect the user’s account.

**3. OPERATOR:**

In this application named as Timezone-Spy, the operator can be employee, student, etc. An operator’s role is to set break i.e. to click on set break button so that the administrator can be notified that the user is on break, end break i.e. to click on end break button so that the administrator can be notified that the user has ended break and can enter break reason in order to inform the operator about the reason for the break taken.

**4. REPORT**

Generate reports on defects. Reports can be generated by the administrator.

**5. SEARCH**

Search can be provided by every admin of this application. Administrator can search number of employees. Search each user’s data.

**6. REGISTRATION AND AUTHENTICATION**

Every user is a registered person. This registration can be done by the administrator. After successful completion of registration process a user can able to log in into the system. The user credentials can be validated in this authentication module.

**1.3.2TECHNICAL REQUIREMENT**

**HARDWARE REQUIREMENTS:**

* PIV 2.8 GHz Processor and Above
* RAM 512MB and Above
* HDD 20 GB Hard Disk Space and Above

**SOFTWARE REQUIREMENTS:**

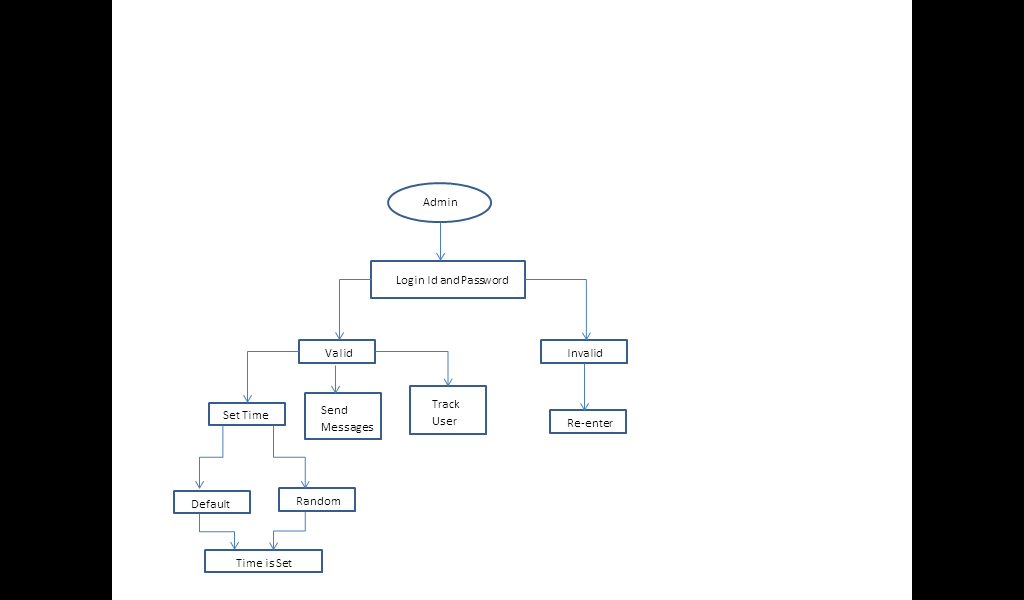
* WINDOWS OS (XP / 7/8)
* Visual Studio 2010 Ultimate Edition
* Visual Studio .Net Framework (Minimal for Deployment)
* SQL Server 2008 Enterprise Edition.
* Microsoft SQL Server R2 Pack((Minimal for Deployment

## 1.3.3 SOFTWARE REQUIREMENT SPECIFICATION

The following are the requirements that have been specified for falling into broad categories:

Software Specifications:-

* Operating System : Windows XP or Windows 7
* Front-end : VB.NET
* Back-end : Microsoft SQL
* Database language : vb.net

**1.3.4 REQUIREMENT DECOMPOSITION DIAGRAM:**

**1.4Tools and Technology:**

1. **VB.NET**

## SERVER APPLICATION DEVELOPMENT;

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

**SERVER-SIDE MANAGED CODE:**

VB.NET is the hosting environment that enables developers to use the .NET Framework to target Windows-based applications. However, VB.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Windows Forms use VB.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of windows based applications. The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

If you develop and publish your own windows application, the .NET Framework provides a set of classes that conform to all the underlying communication standards. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

Finally, like Windows Forms application in the managed environment, it will run with the speed of native machine language using the scalable communication of SQL.

**ACTIVE SERVER VB.NET:**

VB.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Windows applications. VB.NET offers several important advantages over previous Windows development models:

* **Enhanced Performance.** VB.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, VB.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
* **World-Class Tool Support.** The VB.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment.
* **Power and Flexibility.** Because VB.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Window application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Windows. VB.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages.
* **Simplicity.** VB.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the VB.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.
* **Manageability.** VB.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Windows applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying VB.NET Framework applications as well. A VB.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.
* **Scalability and Availability.** VB.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the VB.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.
* **Security.** With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

# **INTRODUCTION TO .NET Framework:**

# The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Windows-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and Remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

**MICROSOFT WORD :**

**Microsoft Word 2010** (often called Word) is a graphical word processing program that users can type with. It is made by the computer company Microsoft. The purpose of the MS Word is to allow the users to type and save documents.

Similar to other word processors, it has helpful tools to make documents paper. Some of the important tools are -:

* A spelling & grammar checker, word count (this also counts letters and lines).
* Includes speech recognition (a technology that lets users control their computers by speaking to it, or telling it what to write).

We can make attractive documents, insert pictures in documents, make web pages, graphs etc. Also you can create tables. Also, it displays synonyms (similar words) of words and can read out the text. It also can print in different ways. It gives different page layouts. Also allows to Draw Charts, Smart Arts. It gives multiple Font option with variable size

**Use of Microsoft Word 20010 in development of project**

Using some of the features of Microsoft Word 2010. The content for the black book is been typed which means a soft copy of BLACK BOOK which contain project details has been created. Which also helped to provide a constant Font Type and size used in soft copy is Font Type is Times New Roman with size of 14 or 12 and Border for the page Layout

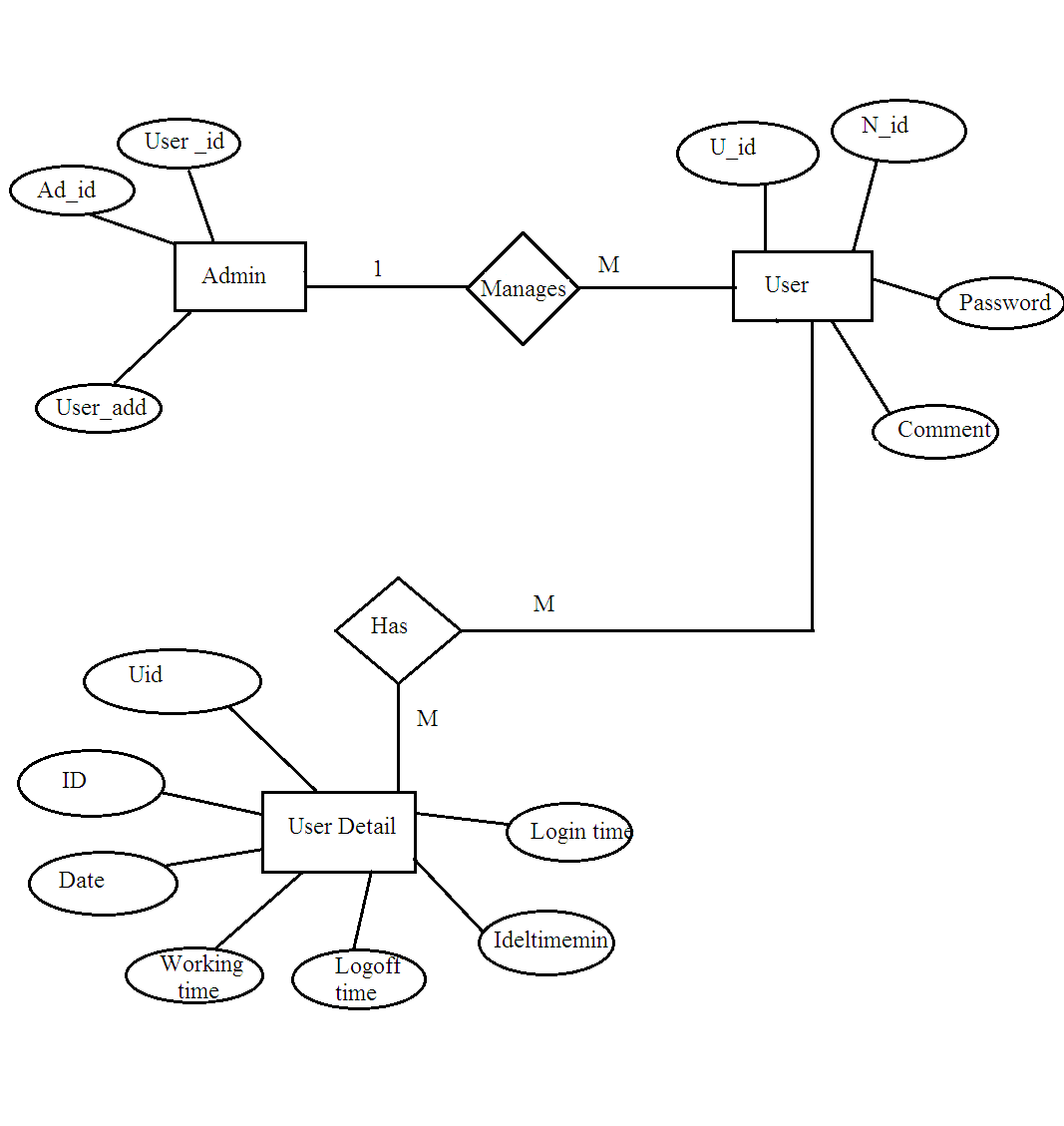
**2.DESIGN PHASE**

* 1. **DETAILED LIFECYCLE OF PROJECT(LOGICAL DESIGN)**
     1. **E-R Diagram:**
  + The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
  + The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
  + The set of primary components that are identified by the ERD are

◆Data object ◆Relationships

◆Attributes ◆Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.

****

* + 1. **Context Diagram :**

User

Login Data

Accept Registration

Fetch Data

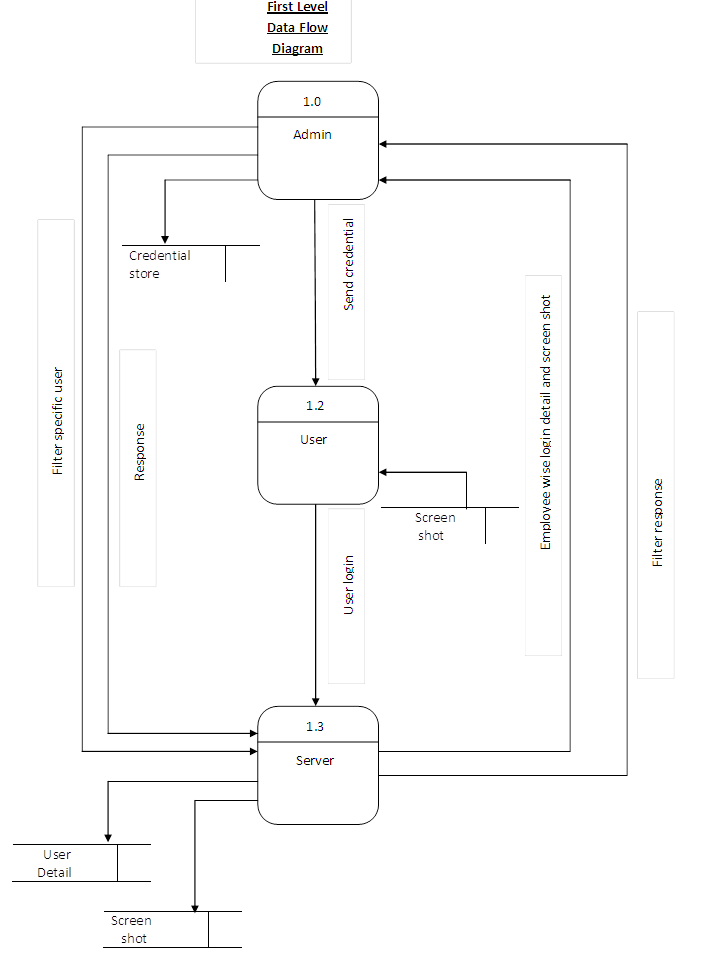
Track User

Admin

**2.1.3 EVENT TABLE**

****

* + 1. **DFD:2.1.3.1 DFD Level 0, Level 1, and so on**

****

**2.2 Physical Design**

**2.2.1 System Flow Chart**

**START**

**Login**

**Enter User Name and password**

**Valid**

NO

Yes

**Login the application**

**Users**

**Operators**

**Start Application**

**Set break**

**End break**

**Reason for**

**STOP**

**Send message**

**Admin**

**2 Implementation Phase**

**2.1 Coding/Building**

**Admin Panel**

**User.vb**

Imports System.Data.SqlClient

Imports System.Security.Cryptography

Imports System.Text

PublicClassUser

Dim myconnection AsSqlConnection

Dim mycommand AsSqlCommand

Dim ds AsDataSet

Dim da AsSqlDataAdapter

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

da = NewSqlDataAdapter("select \* from users where userName like '"& txtuserName.Text &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

MessageBox.Show("User already exist try another name", "Duplication Violation", MessageBoxButtons.OK)

Else

If (txtPassword.Text.Equals(txtRetypePass.Text)) Then

mycommand = NewSqlCommand("insert into users ([UserName],[Password],[active]) values ('"& txtuserName.Text &"','"& GetSHA1HashData(txtPassword.Text) &"','"&"active"&"')", myconnection)

mycommand.ExecuteNonQuery ()

txtPassword.Text = ""

txtRetypePass.Text = ""

txtuserName.Text = ""

MessageBox.Show ("New User Created Successfully", "User Created", MessageBoxButtons.OK)

Me.Hide()

Else

MessageBox.Show ("password and Re-type password are not same", "Password Violation", MessageBoxButtons.OK)

EndIf

EndIf

EndSub

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

Myconnection = NewSqlConnection ("Data Source=ACER-PC;AttachDbFilename=C:\eoy\ideltime.mdf;Integrated Security=True; User ID=administrator; Password=1234; Encrypt=False ;")

myconnection.Open()

EndSub

PrivateFunction GetSHA1HashData(ByVal data AsString) AsString

'create new instance of md5

Dim sha1 AsSHA1 = sha1.Create()

'Convert the input text to array of bytes

Dim hash Data () As Byte = sha1.ComputeHash(Encoding.Default.GetBytes(data))

'create new instance of StringBuilder to save hashed data

Dim ReturnValue AsStringBuilder = NewStringBuilder()

'Loop for each byte and add it to StringBuilder

Dim i AsInteger

For i = 0 To hashData.Length - 1 Step i + 1

ReturnValue. Append (hashData(i).ToString ())

Next

' MessageBox.Show(ReturnValue.ToString(), "status")

' return hexadecimal string

Return ReturnValue.ToString()

EndFunction

EndClass

**Form1.vb**

Imports System.Data.SqlClient

PublicClassForm1

Dim myconnection As SqlConnection

Dim mycommand As SqlCommand

Dim da As SqlDataAdapter

Dim ds, dsb As DataSet

Dim tim () AsString

Dim tempid AsString

Dim user, dateSel AsString

Dim format1 AsString = "MM d yyyy"

PrivateSub Form1\_Load (ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

'TODO: This line of code loads data into the 'IDELTIMEDataSet1.breakTable' table. You can move, or remove it, as needed.

'Me.BreakTableTableAdapter.Fill(Me.IDELTIMEDataSet1.breakTable)

'TODO: This line of code loads data into the 'IdeltimeDataSet.usertable' table. You can move, or remove it, as needed.

'Smyconnection = New SqlConnection ("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\admin\Downloads\SystemIdleTimerTest\_src2 (1)\SystemIdleTimerTest\_src\SystemIdleTimerTest\SystemIdleTimer\ideltime.mdf;Integrated Security=True; User Instance=True")

Myconnection = New SqlConnection ("Data Source=ACER-PC;AttachDbFilename=C:\eoy\ideltime.mdf;Integrated Security=True")

myconnection.Open ()

da = New SqlDataAdapter("select distinct userName from user Table", myconnection)

ds = New DataSet()

da.Fill(ds, "Pdetail")

dgUser.DataSource = ds

dgUser.DataMember = "Pdetail"

EndSub

PrivateSub dgUser\_CellContentClick (ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles dgUser.CellContentClick

lblsDate.Text = ""

lblLoginTime.Text = ""

lblIdelTime.Text = ""

lblWorkingTime.Text = ""

'MessageBox.Show(e.RowIndex)

user = ""

MessageBox.Show(dgUser.Item(0, e.RowIndex).Value.ToString())

User = dgUser.Item(0, e.RowIndex).Value.ToString()

lblUser.Text = user

'da = New SqlDataAdapter("select Date from userTable where userName like '" & user & "'", myconnection)

'ds = New DataSet()

'da.Fill(ds)

'For i As Integer = 0 To ds.Tables(0).Rows.Count - 1

'cbDate.Items.Add(ds.Tables(0).Rows(i)(0).ToString())

'Next

EndSub

'Private Sub cbDate\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cccbDate.SelectedIndexChanged

' DateSel = cccbDate.SelectedItem.ToString ()

' lblsDate.Text = cccbDate.SelectedItem.ToString () & “(mm dd yyyy) format"

' da = New SqlDataAdapter("select \* from userTable where userName like '" & user & "' and Date like '" & dateSel & "'", myconnection)

' ds = New DataSet()

' da.Fill(ds)

' lblLoginTime.Text = ds.Tables(0).Rows(0)(4).ToString() & " (24 hour format)"

' lblIdelTime.Text = ds.Tables(0).Rows(0)(3).ToString() & " (min)"

' lblWorkingTime.Text = ds.Tables(0).Rows(0)(6).ToString()

'End Sub

PrivateSub cbDate\_ValueChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cbDate.ValueChanged

dateSel = cbDate.Text

lblsDate.Text = cbDate.Text.ToString() &" (mm dd yyyy) format"

da = New SqlDataAdapter("select \* from userTable where userName like '"& user &"' and Date like '"& dateSel &"'", myconnection)

ds = New DataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

tim = ds.Tables(0).Rows(0)(4).ToString().Split("|")

lblLoginTime.Text = tim(0) &" (24 hour format)"

lblIdelTime.Text = ds.Tables(0).Rows(0)(3).ToString() &" (min)"

lblWorkingTime.Text = ds.Tables(0).Rows(0)(6).ToString()

tempid = ds.Tables(0).Rows(0)(0).ToString()

da = New SqlDataAdapter ("select startBreak,endBreak,reasonForBreak from breakTable where utid like '"& tempid &"'", myconnection)

dsb = New DataSet()

da.Fill(dsb, "Breakdetail")

dgBreak.DataSource = dsb

dgBreak.DataMember = "Breakdetail"

Else

lblLoginTime.Text = "Not Availabel"

lblIdelTime.Text = "Not Availabel"

lblWorkingTime.Text = "Not Availabel"

dgBreak.DataSource = Nothing

dgBreak.Refresh()

EndIf

EndSub

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

cbDate\_ValueChanged(sender, e)

EndSub

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

End

EndSub

EndClass

**Screen capture**

**Form1.vb**

Option Explicit On

Imports System.Data.SqlClient

Imports Microsoft.VisualBasic

Imports Microsoft.Win32

Imports System.IO

PublicClassForm1

Private Strt As System.Threading.Thread

PublicShared idd AsInteger

Dim myconnection AsSqlConnection

Dim mycommand AsSqlCommand

Dim cmds AsSqlCommand

Dim dr AsSqlDataReader

Dim da AsSqlDataAdapter

Dim ds AsDataSet

Dim itime AsInteger

Dim format1 AsString = "MM d yyyy"

Dim diffh, diffm, difft, hi, ii, hw, iw, temph, tempm, lt, ll, idt, currt, totbreak AsString

Dim ltm(), sbreak(), ebreak(), lts, temp(), a AsString

Dim upd AsString = "add"

Dim tempwait, idstr, lgtm AsString

Dim i1, i2, tb, temptime AsInteger

Dim tempp AsInteger = 0

Dim ss AsString = ""

Dim sk AsString = ""

PublicSubNew()

InitializeComponent()

Icon = My.Resources.TrayIcon

EndSub

PrivateSub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

Try

con()

Dim lt AsString

da = NewSqlDataAdapter("select logofftime from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

lt = ""

lt = ds.Tables(0).Rows(0)(0).ToString()

If lt.Equals("") Then

Else

lt = lt + "|"

EndIf

If (TimeOfDay.Hour < 10) Then

temph = "0"& TimeOfDay.Hour

Else

temph = TimeOfDay.Hour

EndIf

If (TimeOfDay.Minute < 10) Then

tempm = "0"& TimeOfDay.Minute

Else

tempm = TimeOfDay.Minute

EndIf

mycommand = NewSqlCommand("update usertable set logofftime = '"& temph &":"& tempm &"' where id like '"&Form3.uid &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

mycommand.ExecuteNonQuery()

Timer3.Enabled = True

Form3.ststus = "loginHide"

Form3.txtUserName.Text = ""

Form3.txtPassword.Text = ""

EndIf

Catch ex AsException

EndTry

EndSub

Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load

' Me.BackColor = Color.FromArgb(152, 0, 136)

' Me.TransparencyKey = Color.FromArgb(152, 0, 136)

If Form3.ststus.Equals("loginHide") Then

Form3.txtPassword.Text = ""

Form3.txtUserName.Text = ""

Timer3.Enabled = True

Else

Try

'Shell("taskmgr.exe", AppWinStyle.Hide)

myconnection = NewSqlConnection("Data Source=SERVER\SQLEXPRESS;Initial Catalog=IDELTIME;Persist Security Info=True;User ID=sa;Password=sp@1234;Encrypt=False;")

myconnection.Open()

Ref()

Me.Top = 2

Me.Left = 2

If SystemIdleTimer1.IsRunning = FalseThen

SystemIdleTimer1.MaxIdleTime = CUInt(300)

SystemIdleTimer1.Start()

Else

SystemIdleTimer1.Stop()

EndIf

Dim Generator As System.Random = New System.Random()

temptime = Convert.ToInt32(Generator.Next(10, 15))

da = NewSqlDataAdapter("select \* from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

idd = ds.Tables(0).Rows(0)(0)

If ds.Tables(0).Rows.Count <= 0 Then

If (TimeOfDay.Hour < 10) Then

temph = "0"& TimeOfDay.Hour

Else

temph = TimeOfDay.Hour

EndIf

If (TimeOfDay.Minute < 10) Then

tempm = "0"& TimeOfDay.Minute

Else

tempm = TimeOfDay.Minute

EndIf

mycommand = NewSqlCommand("insert into usertable([userName],[Date],[ideltimeMin],[logintime],[logofftime],[ipaddress]) values ('"&Form3.UserNm &"','"&DateTime.Now.ToString(format1) &"','"&"0"&"','"& temph &":"& tempm &"','"&""&"','15')", myconnection)

mycommand.ExecuteNonQuery()

mycommand = NewSqlCommand("insert into UniqueUsertable([userName],[Date],[ideltimeMin],[logintime],[logofftime],[ipaddress]) values ('"&Form3.UserNm &"','"&DateTime.Now.ToString(format1) &"','"&"0"&"','"& temph &":"& tempm &"','"&""&"','15')", myconnection)

mycommand.ExecuteNonQuery()

Else

ll = ""

ll = ds.Tables(0).Rows(0)(4).ToString()

If (TimeOfDay.Hour < 10) Then

temph = "0"& TimeOfDay.Hour

Else

temph = TimeOfDay.Hour

EndIf

If (TimeOfDay.Minute < 10) Then

tempm = "0"& TimeOfDay.Minute

Else

tempm = TimeOfDay.Minute

EndIf

'mycommand = New SqlCommand("update usertable set logintime = '" & ll & "|" & temph & ":" & tempm & "' where userName like '" & Environment.UserName.ToString() & "\_" & Environment.MachineName & "' and Date like '" & DateTime.Now.ToString(format1) & "'", myconnection)

'mycommand.ExecuteNonQuery()

EndIf

Ref()

Catch ex AsException

EndTry

EndIf

EndSub

PrivateSub SystemIdleTimer1\_OnEnterIdleState(ByVal sender As System.Object, ByVal e As EdinDazdarevic.IdleEventArgs) Handles SystemIdleTimer1.OnEnterIdleState

'MessageBox.Show("Entered idle state")

hi = TimeOfDay.Hour

ii = TimeOfDay.Minute

EndSub

PrivateSub SystemIdleTimer1\_OnExitIdleState(ByVal sender As System.Object, ByVal e As EdinDazdarevic.IdleEventArgs) Handles SystemIdleTimer1.OnExitIdleState

'MessageBox.Show("Welcome back!")

Try

Dim da1 AsSqlDataAdapter

con()

hw = TimeOfDay.Hour

iw = TimeOfDay.Minute

diffh = Val(hw) - Val(hi)

diffm = Val(iw) - Val(ii)

difft = (diffh \* 60) + diffm

da1 = NewSqlDataAdapter("select id,ideltimeMin from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da1.Fill(ds)

idd = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(0)

itime = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(1)

mycommand = NewSqlCommand("update Usertable set ideltimeMin = '"& Val(itime + difft + 5) &"' where id like '"& idd &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

mycommand.ExecuteNonQuery()

da1 = NewSqlDataAdapter("select ideltimeMin from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da1.Fill(ds)

itime = 0

For intLoopIndex = 0 To ds.Tables(0).Rows.Count - 1

itime = itime + Val(ds.Tables(0).Rows(0)(0).ToString())

Next intLoopIndex

mycommand = NewSqlCommand("update UniqueUsertable set ideltimeMin = '"& Val(itime) &"' where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

mycommand.ExecuteNonQuery()

'lblIt.Text = Val(itime + difft + 5)

'itime =

'If upd <> "" Then

' upd = ""

'End If

'Ref()

Strt = New System.Threading.Thread(AddressOf MyThread1)

Strt.Start()

Catch ex AsException

EndTry

EndSub

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

Form3.ststus = "Hide"

Me.DialogResult = Windows.Forms.DialogResult.Ignore

Me.Hide()

EndSub

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

Form2.Show()

Form2.BreakReason.Text = ""

da = NewSqlDataAdapter("select id from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

idd = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(0)

'Ref()

Strt = New System.Threading.Thread(AddressOf MyThread1)

Strt.Start()

EndSub

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

Try

con()

btnBreak.Visible = True

btnContinue.Visible = False

'MessageBox.Show("break over")

ll = ""

da = NewSqlDataAdapter("select max(id) from breakTable group by utid having utid like '"& idd &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

ll = ds.Tables(0).Rows(0)(0).ToString()

'If ll.Equals("") Then

'Else

' ' ll = ll + "|"

'End If

If (TimeOfDay.Hour < 10) Then

temph = "0"& TimeOfDay.Hour

Else

temph = TimeOfDay.Hour

EndIf

If (TimeOfDay.Minute < 10) Then

tempm = "0"& TimeOfDay.Minute

Else

tempm = TimeOfDay.Minute

EndIf

mycommand = NewSqlCommand("update breakTable set endBreak = '"& temph &":"& tempm &"' where id like '"& ll &"'", myconnection)

mycommand.ExecuteNonQuery()

'mycommand = New SqlCommand("update breakTable set reasonForBreak = '" & Form2.reason & "' where id like '" & idd & "'", myconnection)

'mycommand.ExecuteNonQuery()

EndIf

'Else

'Form2.Show()

''Form3.ststus = "Continue"

'Form2.BreakReason.Text = ""

'End If

da = NewSqlDataAdapter("select id from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

idd = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(0)

'Ref()

Strt = New System.Threading.Thread(AddressOf MyThread1)

Strt.Start()

'btnBreak.Visible = True

'btnContinue.Visible = False

Catch ex AsException

EndTry

EndSub

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

Ref()

EndSub

Sub MyThread1()

Ref()

EndSub

PublicSub Ref()

'Me.BackColor = Color.FromArgb(152, 0, 136)

'Me.TransparencyKey = Color.FromArgb(152, 0, 136)

Try

con()

IfMe.InvokeRequired Then

'Me.Invoke(New MethodInvoker(AddressOf Ref))

Else

da = NewSqlDataAdapter("select \* from usertable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

a = "|"

ll = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(4).ToString()

lts = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(5).ToString()

'idt = ds.Tables(0).Rows(0)(3).ToString()

ltm = lts.Split(a)

temp = ll.Split(a)

currt = TimeOfDay.Hour &":"& TimeOfDay.Minute

Dim aa(), bb() AsString

aa = temp(0).Split(":")

bb = currt.Split(":")

i1 = Val(bb(0) - aa(0)) \* 60

i2 = Val(bb(1)) - Val(aa(1))

lgtm = ds.Tables(0).Rows(0)(4).ToString()

loginTime.Text = "Login time : "& ds.Tables(0).Rows(0)(4).ToString()

tempwait = ""&Math.Floor((i1 + i2 - idt) / 60).ToString() &" hour "&CInt((i1 + i2 - idt) Mod 60).ToString() &" min"

'lblwtime.Text = "" & Math.Floor((i1 + i2 - idt) / 60).ToString() & " hour " & CInt((i1 + i2 - idt) Mod 60).ToString() & " min"

mycommand = NewSqlCommand("update usertable set workingTime = '"& tempwait &"' where id like '"&Form3.uid &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

mycommand.ExecuteNonQuery()

'da = New SqlDataAdapter("select ideltimeMin from userTable where userName like '" & Form3.UserNm & "' and Date like '" & DateTime.Now.ToString(format1) & "'", myconnection)

'ds = New DataSet()

'da.Fill(ds)

'itime = 0

'For intLoopIndex = 0 To ds.Tables(0).Rows.Count - 1

' itime = Val(itime) + Val(ds.Tables(0).Rows(intLoopIndex)(0).ToString())

'Next intLoopIndex

'lblIt.Text = itime & " min"

da = NewSqlDataAdapter("select workingTime from userTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

Dim intLoopIndex, toth, totm AsInteger

toth = 0

totm = 0

Dim stt AsString

Dim totalw() AsString

For intLoopIndex = 0 To ds.Tables(0).Rows.Count - 1

stt = ds.Tables(0).Rows(intLoopIndex)(0).ToString()

totalw = stt.Split(" ")

toth = toth + Val(totalw(0))

totm = totm + Val(totalw(2))

Next intLoopIndex

If totm > 60 Then

totm = totm - Val(lblIt.Text)

toth = toth + (totm / 60)

totm = totm Mod 60

EndIf

lblwtime.Text = toth &" hour "& totm &" min"

EndIf

mycommand = NewSqlCommand("update uniqueuserTable set workingTime = '"& lblwtime.Text &"' where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

mycommand.ExecuteNonQuery()

da = NewSqlDataAdapter("select id from UserTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

'write logic of idstr with all id should transfer in query: (,) logic

For intLoopIndex = 0 To ds.Tables(0).Rows.Count - 1

If (intLoopIndex = 0) Then

idstr = ds.Tables(0).Rows(intLoopIndex)(0)

Else

idstr = idstr &","& ds.Tables(0).Rows(intLoopIndex)(0)

EndIf

Next intLoopIndex

da = NewSqlDataAdapter("select ideltimeMin from UserTable where userName like '"&Form3.UserNm &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

Dim tmpidel AsInteger

If ds.Tables(0).Rows.Count > 0 Then

For intLoopIndex = 0 To ds.Tables(0).Rows.Count - 1

tmpidel = tmpidel + Val(ds.Tables(0).Rows(intLoopIndex)(0).ToString())

Next

lblIt.Text = tmpidel.ToString()

EndIf

da = NewSqlDataAdapter("select \* from breakTable where utid in ("& idstr &")", myconnection)

ds = NewDataSet()

da.Fill(ds)

totbreak = "0"

ForMe.tb = 0 To ds.Tables(0).Rows.Count - 1

sbreak = ds.Tables(0).Rows(tb)(2).ToString().Split(":")

ebreak = ds.Tables(0).Rows(tb)(3).ToString().Split(":")

If (ebreak(0) = "") Then

diffh = Val(TimeOfDay.Hour) - Val(sbreak(0))

diffm = Val(TimeOfDay.Minute) - Val(sbreak(1))

Else

diffh = Val(ebreak(0)) - Val(sbreak(0))

diffm = Val(ebreak(1)) - Val(sbreak(1))

EndIf

'diffh = Val(ebreak(0)) - Val(sbreak(0))

'diffm = Val(ebreak(1)) - Val(sbreak(1))

totbreak = Val(totbreak) + (Val(diffh) \* 60) + Val(diffm)

Next

lblBreakTime.Text = ""&Math.Floor(totbreak / 60).ToString() &" hour "&CInt(totbreak Mod 60).ToString() &" min"

'lblBreakTime.Text = "" & totbreak & " (min)"

EndIf

da = NewSqlDataAdapter("select \* from NewsTable ", myconnection)

ds = NewDataSet()

da.Fill(ds)

lblNews.Text = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(1).ToString()

Catch ex AsException

EndTry

EndSub

Sub con()

Try

If (myconnection.State = ConnectionState.Closed) Then

myconnection.Open()

EndIf

Catch ex AsException

EndTry

EndSub

PrivateSub Timer2\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer2.Tick

If sk.Equals("nd") Then

Else

Try

con()

upd = "add"

da = NewSqlDataAdapter("select \* from usertable where id like '"&Form3.uid &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

a = "|"

ll = ds.Tables(0).Rows(0)(4).ToString()

lts = ds.Tables(0).Rows(0)(5).ToString()

idt = ds.Tables(0).Rows(0)(3).ToString()

ss = ds.Tables(0).Rows(0)("ipaddress").ToString()

If ss = ""Then

temptime = 15

'Timer2.Interval = 60000

Else

temptime = Val(ss)

'Timer2.Interval = Val(ss) \* 6000

EndIf

ltm = lts.Split(a)

temp = ll.Split(a)

currt = TimeOfDay.Hour &":"& TimeOfDay.Minute

Dim aa(), bb() AsString

aa = temp(0).Split(":")

bb = currt.Split(":")

i1 = Val(bb(0) - aa(0)) \* 60

i2 = Val(bb(1)) - Val(aa(1))

loginTime.Text = "Login time : "& lgtm

'lblIt.Text = "" & idt & " min"

'lblwtime.Text = "" & Math.Floor((i1 + i2 - idt) / 60).ToString() & " hour " & CInt((i1 + i2 - idt) Mod 60).ToString() & " min"

tempwait = ""&Math.Floor((i1 + i2 - idt) / 60).ToString() &" hour "&CInt((i1 + i2 - idt) Mod 60).ToString() &" min"

'lblttime = "" & Math.Floor((i1 + i2) / 60).ToString() & " hour " & CInt((i1 + i2) Mod 60).ToString() & " min"

mycommand = NewSqlCommand("update usertable set workingTime = '"& tempwait &"',logofftime = '"& currt &"'where id like '"&Form3.uid &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

mycommand.ExecuteNonQuery()

tempp = tempp + 1

'tempp = temptime

If tempp >= temptime Then

Try

tempp = 0

Dim bounds AsRectangle

Dim screenshot As System.Drawing.Bitmap

Dim graph AsGraphics

bounds = Screen.PrimaryScreen.Bounds

screenshot = New System.Drawing.Bitmap(bounds.Width, bounds.Height, System.Drawing.Imaging.PixelFormat.Format32bppArgb)

graph = Graphics.FromImage(screenshot)

graph.CopyFromScreen(bounds.X, bounds.Y, 0, 0, bounds.Size, CopyPixelOperation.SourceCopy)

PictureBox1.Image = screenshot

'Timer1.Enabled = False

Me.Opacity = 100

Dim time AsDateTime = DateTime.Now

Dim format AsString = "MMM ddd d HH mm yyyy"

Dim machinename AsString = Form3.UserNm

If System.IO.Directory.Exists("\\server\DOTNET\Images\"& machinename &"\") Then

PictureBox1.Image.Save("\\server\DOTNET\Images\"& machinename &"\"&Form3.UserNm &"\_"& time.ToString(format) &".gif", System.Drawing.Imaging.ImageFormat.Gif)

Else

System.IO.Directory.CreateDirectory("\\server\DOTNET\Images\"& machinename &"\")

PictureBox1.Image.Save("\\server\DOTNET\Images\"& machinename &"\"&Form3.UserNm &"\_"& time.ToString(format) &".gif", System.Drawing.Imaging.ImageFormat.Gif)

EndIf

con()

Dim cmds AsNewSqlCommand("INSERT INTO Information VALUES(@name,@photo)", myconnection)

cmds.Parameters.AddWithValue("@name", "" + Form3.UserNm &"\_" + time.ToString(format) + ".gif")

Dim ms AsNewMemoryStream()

PictureBox1.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Gif)

Dim data AsByte() = ms.GetBuffer()

Dim p AsNewSqlParameter("@photo", SqlDbType.Image)

p.Value = data

cmds.Parameters.Add(p)

cmds.ExecuteNonQuery()

' MessageBox.Show("Name & Image has been saved", "Save", MessageBoxButtons.OK)

'Label1.Visible = False

'PictureBox1.Image.Save("\\server\DOTNET\" & Environment.UserName.ToString() & "\_" & Environment.MachineName & "\_" & time.ToString(format) & ".gif", System.Drawing.Imaging.ImageFormat.Gif)

Strt = New System.Threading.Thread(AddressOf MyThread1)

Strt.Start()

Catch ex AsException

EndTry

EndIf

Catch ex AsException

EndTry

EndIf

EndSub

'Private Sub Timer1\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

' Form3.Show()

' Timer1.Enabled = False

'End Sub

PrivateSub Timer3\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer3.Tick

Form3.Show()

Timer3.Enabled = False

Me.Hide()

EndSub

PrivateSub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

MessageBox.Show("chears")

sk = "nd"

EndSub

Function cat()

Try

Dim bounds AsRectangle

Dim screenshot As System.Drawing.Bitmap

Dim graph AsGraphics

bounds = Screen.PrimaryScreen.Bounds

screenshot = New System.Drawing.Bitmap(bounds.Width, bounds.Height, System.Drawing.Imaging.PixelFormat.Format32bppArgb)

graph = Graphics.FromImage(screenshot)

graph.CopyFromScreen(bounds.X, bounds.Y, 0, 0, bounds.Size, CopyPixelOperation.SourceCopy)

PictureBox1.Image = screenshot

'Timer1.Enabled = False

Me.Opacity = 100

Dim time AsDateTime = DateTime.Now

Dim format AsString = "MMM ddd d HH mm yyyy"

Dim machinename AsString = Form3.UserNm

If System.IO.Directory.Exists("\\server\DOTNET\Images\"& machinename &"\") Then

PictureBox1.Image.Save("\\server\DOTNET\Images\"& machinename &"\"&Form3.UserNm &"\_"& time.ToString(format) &".gif", System.Drawing.Imaging.ImageFormat.Gif)

Else

System.IO.Directory.CreateDirectory("\\server\DOTNET\Images\"& machinename &"\")

PictureBox1.Image.Save("\\server\DOTNET\Images\"& machinename &"\"&Form3.UserNm &"\_"& time.ToString(format) &".gif", System.Drawing.Imaging.ImageFormat.Gif)

EndIf

con()

Dim cmds AsNewSqlCommand("INSERT INTO Information VALUES(@name,@photo)", myconnection)

cmds.Parameters.AddWithValue("@name", "" + Form3.UserNm &"\_" + time.ToString(format) + ".gif")

Dim ms AsNewMemoryStream()

PictureBox1.Image.Save(ms, System.Drawing.Imaging.ImageFormat.Gif)

Dim data AsByte() = ms.GetBuffer()

Dim p AsNewSqlParameter("@photo", SqlDbType.Image)

p.Value = data

cmds.Parameters.Add(p)

cmds.ExecuteNonQuery()

MessageBox.Show("WelCome to SearchCommunication Pvt. Ltd. ")

' MessageBox.Show("Name & Image has been saved", "Save", MessageBoxButtons.OK)

'Label1.Visible = False

'PictureBox1.Image.Save("\\server\DOTNET\" & Environment.UserName.ToString() & "\_" & Environment.MachineName & "\_" & time.ToString(format) & ".gif", System.Drawing.Imaging.ImageFormat.Gif)

Strt = New System.Threading.Thread(AddressOf MyThread1)

Strt.Start()

Catch ex AsException

EndTry

Return 0

EndFunction

PrivateSub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

sk = ""

MessageBox.Show("Testing")

EndSub

PrivateSub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

cat()

EndSub

EndClass

**AppContext.vb**

PublicClassAppContext

InheritsApplicationContext

#Region" Storage "

PrivateWithEvents Tray AsNotifyIcon

PrivateWithEvents MainMenu AsContextMenuStrip

PrivateWithEvents mnuDisplayForm AsToolStripMenuItem

PrivateWithEvents mnuSep1 AsToolStripSeparator

PrivateWithEvents mnuExit AsToolStripMenuItem

#EndRegion

#Region" Constructor "

PublicSubNew()

'Initialize the menus

mnuDisplayForm = NewToolStripMenuItem("Display form")

mnuSep1 = NewToolStripSeparator()

'mnuExit = New ToolStripMenuItem("Exit")

MainMenu = NewContextMenuStrip

MainMenu.Items.AddRange(NewToolStripItem() {mnuDisplayForm})

'Initialize the tray

Tray = NewNotifyIcon

Tray.Icon = My.Resources.TrayIcon

Tray.ContextMenuStrip = MainMenu

Tray.Text = "Eye On You"

'Display

Tray.Visible = True

EndSub

#EndRegion

#Region" Event handlers "

PrivateSub AppContext\_ThreadExit(ByVal sender AsObject, ByVal e As System.EventArgs) \_

HandlesMe.ThreadExit

'Guarantees that the icon will not linger.

Tray.Visible = False

EndSub

PrivateSub mnuDisplayForm\_Click(ByVal sender AsObject, ByVal e As System.EventArgs) \_

Handles mnuDisplayForm.Click

ShowDialog()

EndSub

PrivateSub mnuExit\_Click(ByVal sender AsObject, ByVal e As System.EventArgs) \_

Handles mnuExit.Click

ExitApplication()

EndSub

PrivateSub Tray\_DoubleClick(ByVal sender AsObject, ByVal e As System.EventArgs) \_

Handles Tray.DoubleClick

ShowDialog()

EndSub

#EndRegion

EndClass

**Form3.vb**

Imports System.Data.SqlClient

Imports System.Security.Cryptography

Imports System.Text

PublicClassForm3

PublicShared ststus, idelt AsString

PublicShared UserNm, uid AsString

Dim myconnection AsSqlConnection

Dim mycommand AsSqlCommand

Dim dr AsSqlDataReader

Dim da AsSqlDataAdapter

Dim ds AsDataSet

Dim itime AsInteger

Dim format1 AsString = "MM d yyyy"

Dim passenc, tUser AsString

Dim hi, ii, hw, iw, diffh, diffm, difft, lt, ll, temph, tempm AsString

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

Me.BackColor = Color.FromArgb(152, 0, 136)

Me.TransparencyKey = Color.FromArgb(152, 0, 136)

'btnClose.BackColor = Color.FromArgb(152, 0, 136)

' btnClose.TransparencyKey = Color.FromArgb(152, 0, 136)

'Me.BackColor = Color.Transparent

Form1.btnBreak.Visible = True

Form1.btnContinue.Visible = False

'myconnection = New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\admin\Downloads\SystemIdleTimerTest\_src2(1)\SystemIdleTimerTest\_src\SystemIdleTimerTest\SystemIdleTimer\ideltime.mdf;Integrated Security=True;User Instance=True")

'myconnection = New SqlConnection("Data Source=SPSERVER\SQLEXPRESS;Initial Catalog=IDELTIME;User ID=ashish;Password=aa@1234;Encrypt=False;")

myconnection = NewSqlConnection("Data Source=ACER-PC;AttachDbFilename=C:\eoy\ideltime.mdf;Integrated Security=True;User ID=administrator;Password=1234;Encrypt=False;")

myconnection.Open()

Try

Application.Run(NewAppContext)

Catch ex AsException

EndTry

txtUserName.Text = ""

txtPassword.Text = ""

EndSub

PrivateSub Timer1\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

Timer1.Enabled = False

Me.Hide()

If (myconnection.State = ConnectionState.Closed) Then

myconnection.Open()

EndIf

If UserNm.Equals("") Then

tUser = ""

Else

tUser = UserNm

EndIf

'da = New SqlDataAdapter("select \* from userTable where userName like '" & tUser & "' and Date like '" & DateTime.Now.ToString(format1) & "'", myconnection)

'ds = New DataSet()

'da.Fill(ds)

'If ds.Tables(0).Rows.Count <= 0 Then

If (TimeOfDay.Hour < 10) Then

temph = "0"& TimeOfDay.Hour

Else

temph = TimeOfDay.Hour

EndIf

If (TimeOfDay.Minute < 10) Then

tempm = "0"& TimeOfDay.Minute

Else

tempm = TimeOfDay.Minute

EndIf

mycommand = NewSqlCommand("insert into usertable([userName],[Date],[ideltimeMin],[logintime],[logofftime],[ipaddress]) values ('"& tUser &"','"&DateTime.Now.ToString(format1) &"','"&"0"&"','"& temph &":"& tempm &"','"& temph &":"& tempm &"','15')", myconnection)

mycommand.ExecuteNonQuery()

da = NewSqlDataAdapter("select \* from UniqueuserTable where userName like '"& tUser &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count <= 0 Then

mycommand = NewSqlCommand("insert into UniqueUsertable([userName],[Date],[ideltimeMin],[logintime],[logofftime],[ipaddress]) values ('"& tUser &"','"&DateTime.Now.ToString(format1) &"','"&"0"&"','"& temph &":"& tempm &"','"&""&"','15')", myconnection)

mycommand.ExecuteNonQuery()

'da = New SqlDataAdapter("select ideltimeMin from userTable where userName like '" & UserNm & "' and Date like '" & DateTime.Now.ToString(format1) & "'", myconnection)

'ds = New DataSet()

'da.Fill(ds)

'itime = 0

'For intLoopIndex = 0 To ds.Tables(0).Rows.Count - 1

' itime = Val(itime) + Val(ds.Tables(0).Rows(intLoopIndex)(0).ToString())

'Next intLoopIndex

'idelt = itime

'Form1.lblIt.Text = idelt & " min"

EndIf

da = NewSqlDataAdapter("select \* from userTable where userName like '"& tUser &"' and Date like '"&DateTime.Now.ToString(format1) &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

uid = ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(0).ToString()

EndIf

' ll = ds.Tables(0).Rows(0)(0).ToString()

'Else

'll = ""

'da = New SqlDataAdapter("select logintime from userTable where userName like '" & tUser & "' and Date like '" & DateTime.Now.ToString(format1) & "'", myconnection)

'ds = New DataSet()

'da.Fill(ds)

'If ds.Tables(0).Rows.Count > 0 Then

' ll = ds.Tables(0).Rows(0)(0).ToString()

' If (TimeOfDay.Hour < 10) Then

' temph = "0" & TimeOfDay.Hour

' Else

' temph = TimeOfDay.Hour

' End If

' If (TimeOfDay.Minute < 10) Then

' tempm = "0" & TimeOfDay.Minute

' Else

' tempm = TimeOfDay.Minute

' End If

' mycommand = New SqlCommand("update usertable set logintime = '" & ll & "|" & temph & ":" & tempm & "' where userName like '" & tUser & "' and Date like '" & DateTime.Now.ToString(format1) & "'", myconnection)

' mycommand.ExecuteNonQuery()

'End If

'End If

Form1.Show()

Form1.Ref()

EndSub

PrivateSub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

passenc = GetSHA1HashData(txtPassword.Text)

da = NewSqlDataAdapter("select username,password,active from users where userName like '"& txtUserName.Text &"' and Password like '"& passenc &"'", myconnection)

ds = NewDataSet()

da.Fill(ds)

If ds.Tables(0).Rows.Count > 0 Then

If ds.Tables(0).Rows(ds.Tables(0).Rows.Count - 1)(2).ToString().Equals("active") Then

UserNm = txtUserName.Text

Timer1.Enabled = True

Form3.ststus = "Hide"

Else

MessageBox.Show("user is Deactivated :-: Contact administrator", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

EndIf

Else

MessageBox.Show("user name and password is incorect", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

EndIf

EndSub

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

'Me.Dispose()

btnHide.PerformClick()

EndSub

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

ststus = "loginHide"

Me.DialogResult = Windows.Forms.DialogResult.Ignore

Me.Hide()

EndSub

PrivateFunction GetSHA1HashData(ByVal data AsString) AsString

'create new instance of md5

Dim sha1 AsSHA1 = sha1.Create()

'convert the input text to array of bytes

Dim hashData() AsByte = sha1.ComputeHash(Encoding.Default.GetBytes(data))

'create new instance of StringBuilder to save hashed data

Dim ReturnValue AsStringBuilder = NewStringBuilder()

'loop for each byte and add it to StringBuilder

Dim i AsInteger

For i = 0 To hashData.Length - 1 Step i + 1

ReturnValue.Append(hashData(i).ToString())

Next

' MessageBox.Show(ReturnValue.ToString(), "status")

' return hexadecimal string

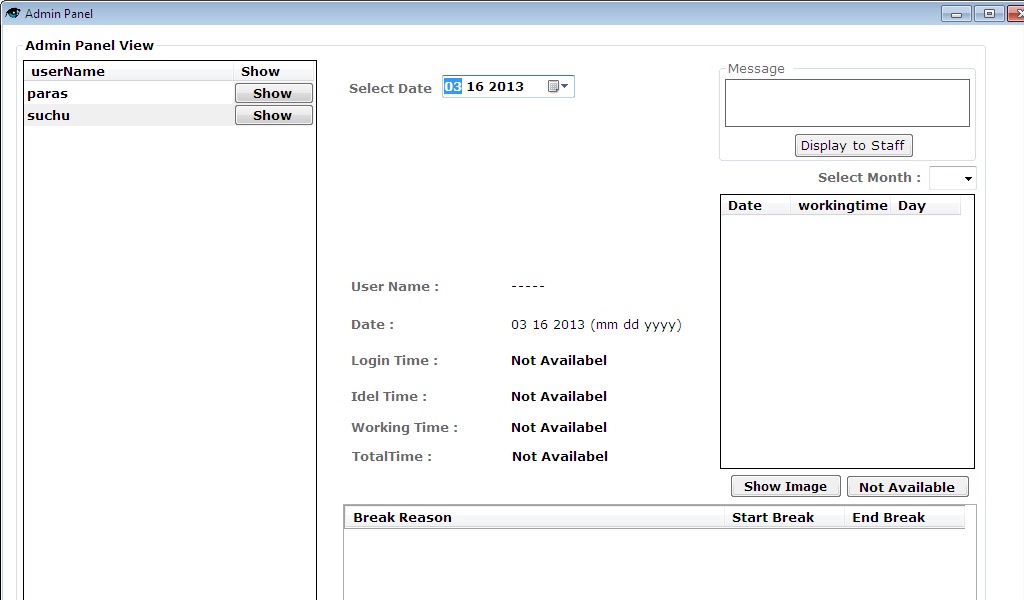
Return ReturnValue.ToString()

EndFunction

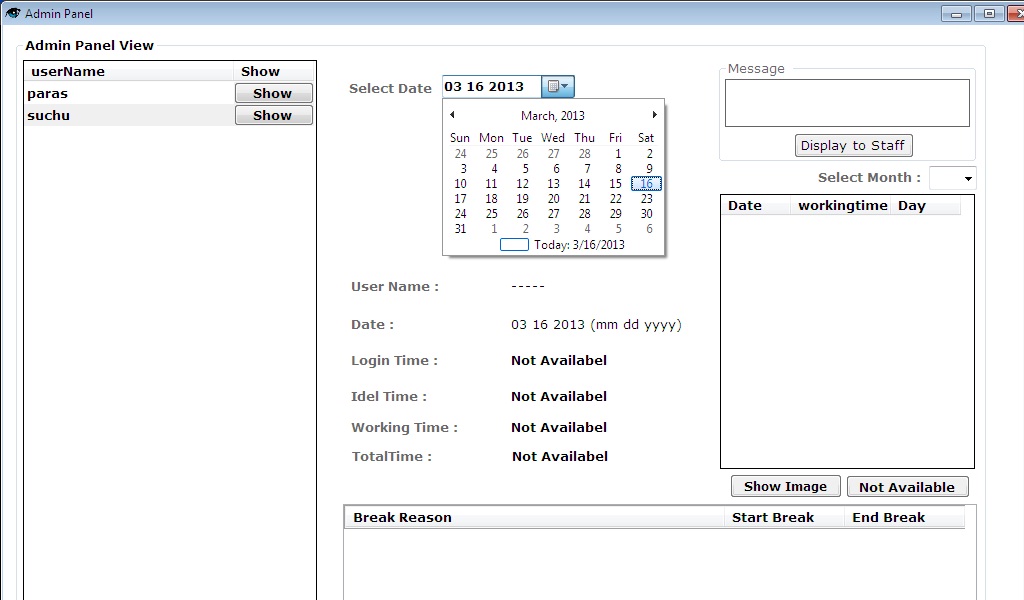
EndClass

* + 1. **SCREEN LAYOUTS:**

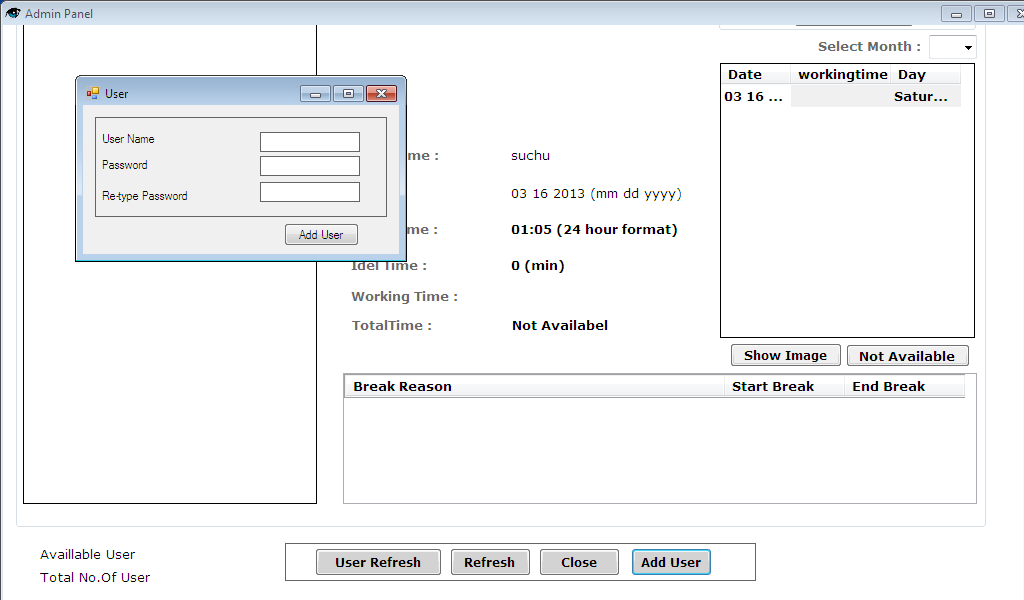
**Admin Panel:**

****

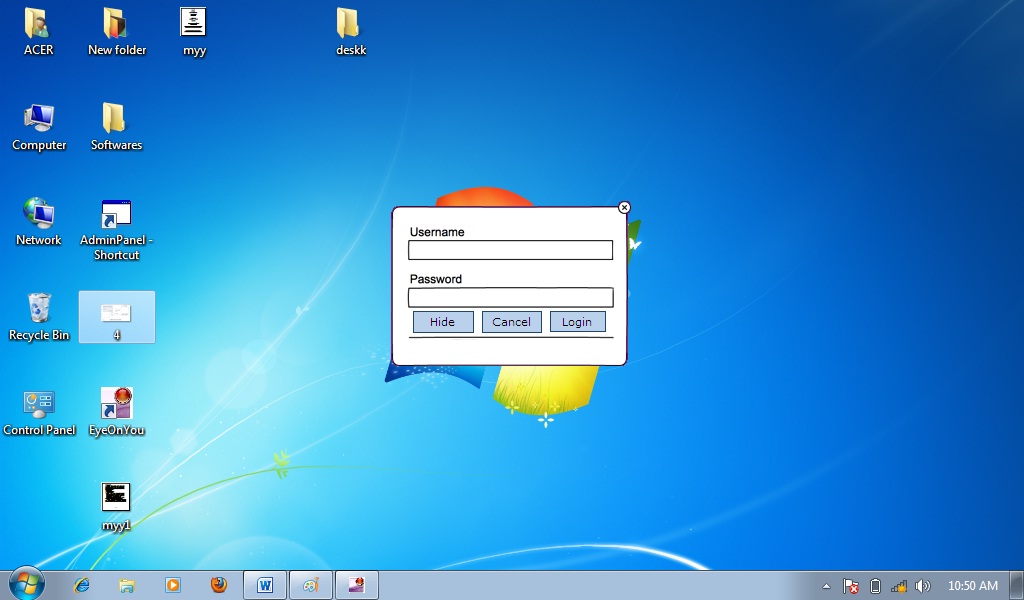
**Select Date**

****

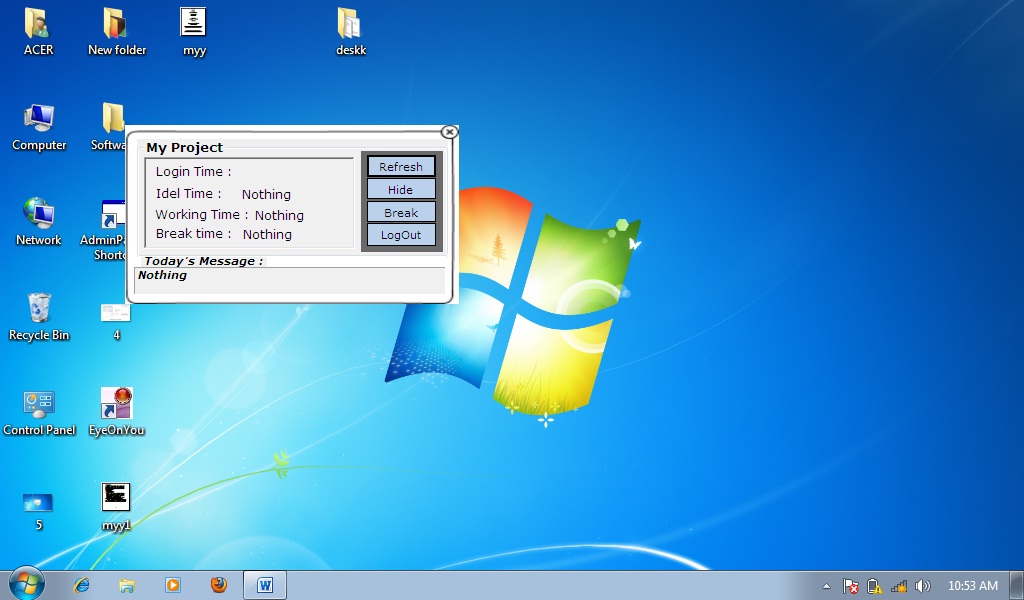
**Add User**



**User Login**

****

**User side interface**

****

* + 1. **Deployment**

**Step1 : Double click on admin\_panel setup file.**

****

**Step 2: Click NEXT to proceed.Then the following screen will appear:**

****

**Step 3:Click NEXT to proceed.Then the following screen will appear:**

****

**Step 4: Click NEXT to proceed.Then the following screen will appear:**



**Step 5:Let the installation get complete.**



**Click on close.**

* 1. **Testing Phase**

**3.2.1 Testing and Its Types**

**Unit Testing:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr no** | **Module** | **Type of testing** | **Why?** | **Desired output** | **Observed output** |
| 1 | Authentication (server) | White and Black box | For checking the input and username and password is getting mathched | Access should be provided only after proper check of field and match of username and password | Access was provided only after proper match of username and password |
| 2 | Retype password | White and Black box | For verifying password | New password should be stored in the database | New password should be stored in the database and is used now onwards |
| 3 | Add user | White and Black box | For adding the user | Records should be added properly | Record is added properly |
| 4 | Send message | White and Black box | For verifying sent message | Message should be sent properly | Message is sent |

**Integration Testing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr no** | **Module** | **Why?** | **Desired output** | **Observed output** |
| 1 | User management module | For testing proper management of employee details | Employee details should be managed with the proper validation | Employee details should be managed with the proper validation |
| 2 | Project management module | Proper management of the project and administration on projects | Projects should be properly added,updated, administered | Projects should be properly added,updated, administered |
| 3 | Assignment management module | Proper management of the project and administration on projects assigned to employee | Projects should be properly added, updated, administered That are assigned to the employees | Projects should be properly added,updated, administered |
| 4 | Performance management | To monitor the performance of the employee | For monitoringreports should be displayed | Reports are being displayed |

**System testing**

In this level of testing we are testing the system as a whole after integrating all the main modules of the project. We are testing whether system is giving correct output or not.

All the modules were integrated and the flows of information among different modules were checked the flow of data is as per the requirements or not .it was also checked that whether any particular module is non –functioning or not i.e. once the integration is over each and every module is functioning in its entirely or not.

In this level of testing we test the following:-

1. Whether all the forms are properly working or not.

2. Whether all the forms are properly linked or not.

3. Whether all the images are properly displayed or not.

4. Whether data retrieval is proper or not.

5. Whether reports are being drawn or not.

6. Whether application is been tracked or not.

**Load/Stress Testing**

**Expected Result**

* Response time should be unaffected irrespective of the no

of users.

* The introduction of the newer clients should not make the

server to work hap hazardously.

* Continuous use of the server by different clients should not result into the server getting slowed down.
* Response time should not be degraded if there is congestion in network.
* If more applications are being tracked simultaneously by a great no of active users then performance should not be degraded of the server.

**Observation**

* The speed of transmission was even when the newer clients were getting added.
* The response of the server was satisfying even with the introduction of newer clients.
* The server continued to work satisfactorily for a good stretch of time also.
* The Server was providing the services properly even if there are more applications for a particular user.

**3.2.2 METHODOLOGIES ADOPTED FOR TESTING**

We have adopted the incremental model while developing the system, while the methodology used is modular approach.

We stared with first developing the database, as it forms a very important part of , then we went on to develop the user interfaces, to decide the overall

Flow of the logic, and thus engage later in integrating the logic.

Its various functions and operations with the user interface.

Once we had managed to develop a standalone tool, we went to write programs to make run on the LAN.

**4Maintenance and Evaluatiion**

**4.1 SYSTEM MAINTENANCE AND FUTURE ENHANCEMENT**

**System Maintenance**

This project is mainly used in organization and the people using it well versed with technology. Hence it is very important that reliable, robust and efficient hardware is used for the product to last longer.

Use of the free software enables easy maintenance of the product.If the form will have syntax error it will not execute the form leasing to the correctness.

Since this project did not involve any budget, it never faced any financial related problems. However, in the future if this project is enhanced to reach at higher level, budget plans have to be passed by the concerned authorities.

The working environment should be clean and virus-free. An anti-virus software, reliable power supply and physical clean workspace will prove to be fruitful for the user.

Training the users to build a skilled workforce is another way to encourage innovation and continuous improvements in performance.

Manuals self-help books can be referred in order to learn its working and functionalities. The provided documentation can lead to better maintenance and lead and future enhancement with the software.

Every System generally has some or other flaws even after undergoing several tasks. Even if Bugs are not present in system, system may not work due to improper configuration. Thus we should take care of future corrections needed ,i.e. post implementation.

**Future Enhancement**

* This application being windows-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
* A Chat application being added.
* The user can ask for permission from the admin and then ask for break.
* The admin can grant permission to the user in the upcoming versions.

**4.2 USER MANUAL**

**Step 1**: If you are admin double click on admin panel.

**Step 2**: If you are user

* **If you are Operator**
* First enter your correct Login id and Password. If your Login id and Password is right then application is opened otherwise retype it again
* On home page you will see some “Menu items” i.e. Refresh, Set break, End break, Log out.
* From **Refresh** you can refresh the application.
* From **Set Break** you can ask for break.
* From **End Break** you can end the break.
* From **Log out**you can Sign out from the application.
* **If you are Administrator**
* First enter your correct Login id and Password. If your Login id and Password is right then your homepage is appeared, otherwise retype it again
* On home page you will see some “Menu items” i.e. Manage User, Manage Components, Defect Management, Reports, Log Out.
* From **Reports** you can do the following task,
* Can see reports.
* Can find any existing user
* Can find Components
* From **Log Out** you can sign out from your System.

**5. ANNEXURE**

**5.1 DATA DICTIONARY**

**5.1.1 Data Name, Aliases if any, length , type**

**TABLE NAME**: **USER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **REQUIRED** | **Description** |
| Primary key | Uid | Int | 4 | Yes | User id |
| Foreign key | Nid | Int | 4 | Yes | News id |
|  | UserName | Nvarchar | 20 | Yes | User name |
|  | Password | Nvarchar | 16 | Yes | User Password |
|  | Active | Nvarchar | 2 | Yes | User active or not |
|  | Comment | Nvarchar | 50 | Yes | User comment |

**TABLE NAME**: **USER DETAIL**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **REQUIRED** | **Description** |
| Primary key | Id | Int | 4 | Yes | User detail id |
| Foreign key | Uid | Int | 4 | Yes | User id |
|  | Date | Date | 8 | Yes | Date |
|  | IdeltimeMin | Nvarchar | 50 | Yes | User idle time |
|  | Logintime | Nvarchar | 50 | Yes | Login time |
|  | Logofftime | Nvarchar | 50 | Yes | Logoff time |
|  | Workingtime | Nvarchar | 50 | Yes | User working time |
|  | Ipaddress | Nvarchar | 50 | Yes | User ip address |

**TABLE NAME**: **UNIQUE USER**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **REQUIRED** | **Description** |
| Primary key | Id | Int | 4 | Yes | User detail id |
| Foreign key | Nid | Int | 4 | Yes | News id |
|  | Username | Nvarchar | 50 | Yes | User name |
|  | Date | Date | 8 | Yes | Date |
|  | IdeltimeMin | Nvarchar | 50 | Yes | User idle time |
|  | Logintime | Nvarchar | 50 | Yes | Login time |
|  | Logofftime | Nvarchar | 50 | Yes | Logoff time |
|  | Workingtime | Nvarchar | 50 | Yes | User working time |
|  | Ipaddress | Nvarchar | 50 | Yes | User ip address |
|  | Nooflogin | Nvarchar | 50 | Yes | User total login |

**TABLE NAME**: **INFORMATION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **REQUIRED** | **Description** |
| Primary key | Id | Int | 4 | Yes | Information id |
| Foreign key | Uid | Int | 4 | Yes | User id |
| Foreign key | Unid | Int | 4 | Yes | Unique user id |
|  | Name | Nvarchar | 50 | Yes | User name |
|  | Screenshort | Image | - | Yes | Screen sort |

**TABLE NAME**: **BREAK TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **REQUIRED** | **Description** |
| Primary key | Id | Int | 4 | Yes | Break id |
| Foreign key | Uid | Int | 4 | Yes | User id |
|  | Startbreak | Nvarchar | 50 | Yes | Break start time |
|  | Endbreak | Nvarchar | 50 | Yes | End break time |
|  | reasonforBreak | Nvarchar | 50 | Yes | Break resion |

**TABLE NAME**: **NEWS TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KEY** | **FIELD NAME** | **DATA TYPE** | **FIELD SIZE** | **REQUIRED** | **Description** |
| Primary key | Nid | Int | 4 | Yes | News id |
|  | News | Nvarchar | 100 | Yes | News |
|  | Newsdate | Nvarchar | 50 | Yes | News date |

**5.3 REFERENCES: BIBLIOGRAPHY, WEBSITES USED**

* **FOR .NET INSTALLATION**

[www.support.mircosoft.com](http://www.support.mircosoft.com)

[www.developer.com](http://www.developer.com)

[www.15seconds.com](http://www.15seconds.com)

* **FOR SQL**

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)

* **FOR .NET**

[www.msdn.microsoft.com/net/quickstart/aspplus/default.com](http://www.msdn.microsoft.com/net/quickstart/aspplus/default.com)

[www.asp.net](http://www.asp.net)

[www.fmexpense.com/quickstart/aspplus/default.com](http://www.fmexpense.com/quickstart/aspplus/default.com)

[www.4guysfromrolla.com/index.aspx](http://www.4guysfromrolla.com/index.aspx)

* **BOOKS**

Spanzars,SAD

* **OTHER WEBSITES**

[www.yahoo.com](file:///C:\Users\ACER\Downloads\www.yahoo.com)