**1. INTRODUCTION TO FITNESS MANAGEMENT SYSTEM :**

Our Fitness Management software is a gym and health club membership management system. we can keep records on your members, their memberships, and have quick and easy communication between you and your members. Fitness Management also includes a booking system, point of sale, accounting and concession has a range of reports that helps in the management of your club.

Our Fitness Management Software is a complete Fitness and recreation facility system program which looks after all of your members, memberships and activities. It is designed for Fitness, recreation centers, and health clubs.

Our Fitness Management Software provides lots of functions such data entry of customer, keeping records of all the things about customer fees, plan, and physical fitness which help to provide good quality of services to customer from Fitness managers.

In the proposed system also provide the total information about equipment and data of Trainers is also stored in it. Services provided by Fitness are also handled by this system.

This system structure is become very simple to understand because of Data Flow Diagram provided by us. Context level Diagram and some chart are also available in this case study. The demo of using the software such as Member detail form, database of software is also provided by us.

**Objective of the Fitness Management System :**

* The main objective of the project is to develop software that facilitates the data storage, data maintenance and its retrieval for the fitness in an igneous way
* To store the record of the customers, the staff that the privileges to access, modify and delete any record and finally the services, Fitness provides to its customers.
* Also, only the staff has the privilege to access any database and make the required changes, if necessary.
* To develop easy-to-use software which handles the customer-staff relationship in an effective manner.
* To develop a user friendly system that requires minimal user training. Most of features and functions are similar to those windows platform.

**Limitation of Existing System :**

* The Existing System such as Workout Trainer is not as much as user (customer) friendly as compare to our proposed System.
* The communication with members is not well in Existing System because all the data is handled by Fitness Manager.
* Customers don’t get full accessibilities to fitness center and all permissions are allowed only for Fitness Manager.
* Today’s system cannot take effort out of finances and debt collection.
* Lot of memory space is required for installing existing software.
* Existing Fitness Management software cannot perform all operation expected by manager such as keeping record of machinery maintenance and service data.

**Abstract of the project Fitness Management System :**

Fitness Management System is an online service that can be setup for your gym to help manage classes, memberships, custom.er managements, surveys and it even has an store the data of customers and trainers.

* It is simple and effective and it’s the way customers want their gym!
* Here is our features list which is continually growing :
* Manage customers
* Manage customers health question forms
* Manage customer surveys
* Screenshot of user statistics graphs
* Manage your trainers and class schedules.
* Class managements
* Create planes & passes

**Some salient features :**

1. Creating & changing Fitness at ease
2. Query Fitness list to any depth
3. Product and component based
4. Simple status and resolutions
5. Multilevel priorities and services

**Modules and description**

1. **Login form :** The fitness Management system first activates the login form. Here the user can enters the user name and password and our system starts the authentication process in with the existing username and password in the database. If the password matches then it allowed to the main page else it warns the user for invalid username and password.
2. **Main form :**

* Membership form : in this form we can enter members details to monitor there data
* Plane form : in this form we are able to offer festive offer to fitness members through providing 50% off on their fees.
* Instructor form : here we can monitor instructors details and maintain their salary.
* Equipment form : here we can monitor the data of equipment.

1. **View**

In this module we can view entire members and instructors fees and salary details and monitor there regular attendance in the fitness management center

**2. SYSTEM ANALYSIS**

**INTRODUCTION TO SYSTEM ANALYSIS**

SYSTEM STUDY AND ANALYSIS :

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and the system developers. System Analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system is identified. The outputs from the organizations are traced to the various processes. System is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optical or at least a satisfactory solution or programing of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. the data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

**2.1 EXISTING SYSTEM**

* Existing system is a manual system.
* Time consuming data entry which includes calculations took a lot of time.
* The proposed system is expected to be faster than the existing system.
* Separate documents are maintained for booking details, hall details and billing details.
* It requires a lot of human involvement and a lot of paperwork needed.

**2.2 PROPOSED SYSTEM**

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces manual work.

* Using this software, it saves the time of the user that he used to spend by doing the work .
* It allows the user to enter the record in a short span of time without any

mistake, if he makes any mistake, he can rectify those mistakes in a short span

of time.

* Proper control of the higher officials.
* Better services
* User friendliness and interactive.
* Minimum time required.
* It allows the user to search a particular record from large amount of records or from the thousand records within a few second
* If any changes are required by the user he can do that in short span of time, without deleting the previous record.
* the user has the advantage of deleting or modifying the record in a short span of time.
* If this software is used then there is no chance of repeating in the consumer number, and serial number.
* This software provides very high security and only authorized people can access the records
* It helps in saving paper time.

**2.3 FEASIBILITY STUDY**

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

**A. TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of

This feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, or running the system once it has been designed.

**Technical Fitness raised during the investigation are :**

* Is the existing technology sufficient for the suggested one?
* Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within the latest technology. Though the technology may become obsolete after some period of time, due to the fact that newer versions of the same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java and the project is technically feasible for development

**B. ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

**The following are some of the important financial questions asked during preliminary**

**Investigation :**

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

**C. BEHAVIORAL FEASIBILITY**

This includes the following questions:

Is there sufficient support for the users?

Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.4 HARDWARE & SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENT

Processor **:** Intel Pentium IV @2.4 GHZ or above

Clock Speed  **:** MHZ

System Bus  **:** 500 Bits

RAM : 1 GB

HDD : 160 GB

Monitor : Plug and play monitor

Keyboard **:** standard 101/102-keys

Mouse **:** PS2

**SOFTWARE REQUIREMENT**

OS : MS WINDOWS 2007

Front End : Microsoft Visual Basic 6.0

Back End : Microsoft Access 2003

**2.5 ABOUT THE SOFTWARE**

MICROSOFT VISUAL BASIC 6.0

Visual Basic 6.0 is Microsoft’s latest version of the Visual Basic Programming language.

Although writing programs can be a tedious chore at times. Visual Basic reduces the effort required on your part, and makes programming enjoyable. Visual Basic makes many aspects of programming as simple as dragging graphics objects onto the screen with your mouse Visual Basic 6.0 is more than just a programming language; the secret to Visual Basic is in its name “Visual”. Within two days of the Windows Operating System, a program must be able to interact with the screen. Keyboard, mouse and printer graphically.

The environment provided by Visual Basic is suitable for any type of application. Using

In this environment, the user can visually design the objects that your application uses. Visual Basic is not just a language, it’s an integrated development environment in which you can develop, run, test and debug your applications. The impressive array of programming resources provided by Visual Basic enables to create the objects extemporaneously which can range from pop-up menu to a message box.

A form is a major part of Visual Basic application, which allows the user to enter the data

As well as view the result. A control is an object that we draw on a form to enable or enhance user interaction with an application. Hence a Visual Basic application is a combination of object-like forms and controls, procedures that can respond to Events and other general-purpose procedures.

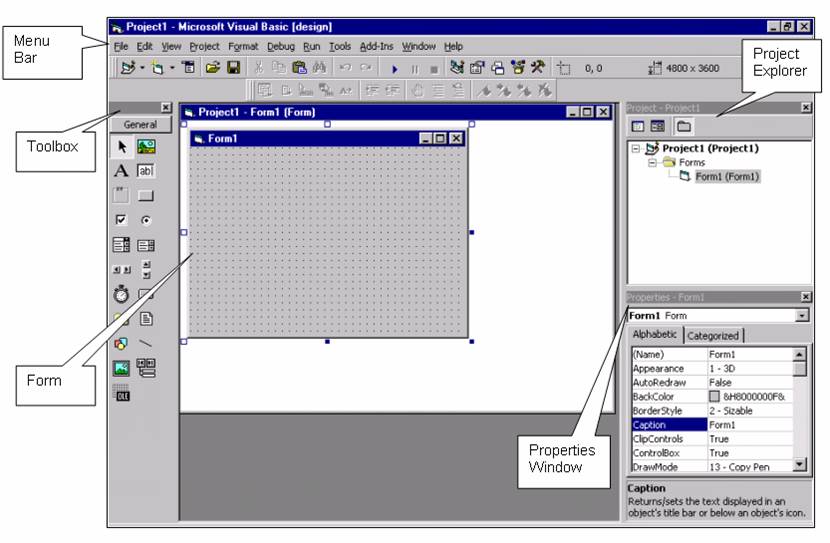
Starting Visual Basic:

Click on the Start button on the Windows taskbar.

Select Programs, then Microsoft Visual Basic

Click on Visual Basic 6.0

**Menu Form :**



At the top of the screen is the Visual Basic Main Window. At the top of the main window is the Title Bar. The title bar gives us information about what program we’re using and

what Visual Basic program we are working with. Under the main menu is the Toolbar. Tittle buttons with pictures also allow us to control Visual Basic, much like the ‘tooltip’ will pop up and tell you what that particular button does.

Parts of a Visual Basic Project.

There are three major components in a Visual Basic project; the project itself, the form, and the controls. Project is the word used to encompass everything in a Visual Basic project.

Other words used to describe a project are application or program. Like he BASIC programming language, Visual Basic was designed for an easy learning curve. Programmers can create both

Simple and complex GUI applications. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions for those components, and writing additional lines of code for more functionality. Since VB defines default attributes and actions for the components, a programmer can develop a simple program

Without writing much code. Programs built with earlier versions suffered performance problems ,but faster computers and native code compilation has made this less of an is, Though VB programs can be compiled into native code executable from version 5 on, they still require the presence of around 1 MB of runtime libraries. Core runtime libraries are included by default

in Windows 2000 and later, but extended runtime components still have to be installed. Earlier versions of Windows (95/98/NT), require that the runtime libraries be distributed with the executable Forms are created using drag-and-drop techniques.

A tool is used to place controls (e.g., text boxes, buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attribute values can be modified during run time based on user actions or changes in the environment, providing a dynamic application. For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands to fill up the form, etc. By inserting code into the event handler a key press in a text box, the program can automatically translate the case of the text being entered, or even prevent certain characters from being inserted Visual Basic can create executable (EXE files), ActiveX controls, or DLL files, but is primarily used to develop Windows applications and to interface database systems. Dialog boxes with less functionality can be used to provide pop-up capabilities. Controls provide the basic functionality of the application, while programmers can insert additional logic within the appropriate event handlers. For example, a drop-down combination box automatically displays a list. When the user selects an element, an event handler is called that executes code that is created to perform the action for that list item. Alternatively, a Visual Basic component can have no user interface, and instead provide ActiveX objects to other programs via Component Object Model (COM). This allows for server-side processing or an add-in module.

The runtime recovers unused memory using reference counting, which depends on variables passing out of scope or being set to nothing, avoiding the problem of memory leaks that are possible in other languages. There is a large library of utility objects, and the language provides basic support for object-oriented programming. Unlike many other programming languages, Visual Basic is generally not case-sensitive—though it transforms keywords into a standard case configuration and forces the case of variable names to conform to the case of the entry in the symbol table. String comparisons are case sensitive by default. The Visual Basic compiler is shared with other Visual Studio languages (C, C++).

Nevertheless, by default the restrictions in the IDE do not allow creation of some targets

(Windows model DLLs) and threading models, but over the years, developers have bypassed these restrictions. Visual Basic is a third-generation event-driven programming language from Microsoft for its Component Object Model (COM) programming model first released in 1991 and declared legacy during 2008. Microsoft intended Visual Basic to be relatively easy to learn and use [1] [2] Visual Basic was derived from BASIC and enables the rapid development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and of ActiveX controls and objects.

A programmer can create an application using the components provided by the Visual Basic program itself. Over time the community of programmers developed third-party components [3][4][5][6][7] Programs written in Visual Basic can also use the Windows API, which requires external function declarations.

The final release was version 6 in 1998 (now known simply as Visual Basic). On April 8,

In 2008, Microsoft stopped supporting Visual Basic 6.0 IDE. The Microsoft Visual Basic team still maintains compatibility for Visual Basic 6.0 applications on Windows Vista, Windows Server 2008 including R2, Windows 7, Windows 8, Windows 8.1, Windows Server 2012, Windows 10, Windows Server 2016, and Windows Server 2019 through its "It Just Works" program [8] In 2014, some software developers still preferred Visual Basic 6.0 over its successor, Visual Basic .NET.

[3][9] In 2014 some developers lobbied for a new version of the VB6 programming

environment.[10][11][12][13] In 2016, Visual Basic 6.0 won the technical impact award at The 19th Annual D.I.C.E. Awards.[14][15][16] A dialect of Visual Basic, Visual Basic for Applications (VBA), is used as a macro or scripting language within several Microsoft , including Microsoft Office.

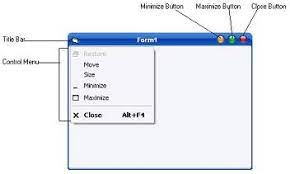
**FORM:**

The form is the most important unit in Visual Basic. It is like a canvas on which the user placed certain objects to develop different interfaces. The form can be of any size and color and the user can attach a code to it which will run when the form is loaded for the first time. A simple project may use only one form while certain applications may require the use of a number of forms. When multiple forms are used each form may represent different operations and outputting results of different operations. Each form is saved on the disk as a separate file with a .FORM extension.

Close

Max button

Minimize button



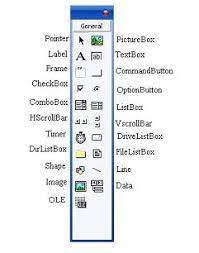
Form 1

Control menu

Tool bar

**CONTROLS:**

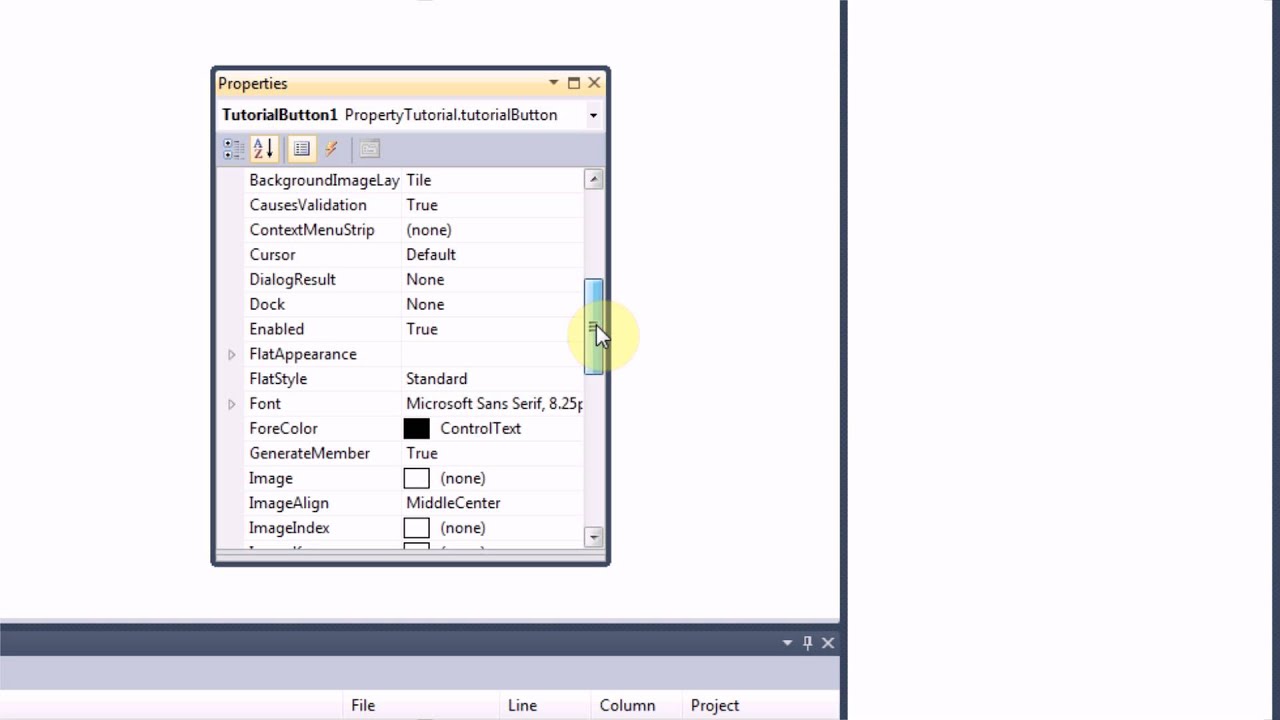
These represent objects or graphical features drawn on forms to allow the user to interact with the system. The objects can be pasted on the form and range from text boxes, labels, scroll bars, command buttons and other menus, to file Management utilities and spreadsheet type grids. Each object may have certain code attached to it, which identifies the operation or activity to be performed by the control. Form and controls together are sometimes called objects.



**PROPERTIES :**

shape and appearance of the object. These characteristics are termed as properties. Some important properties include names, captions, size, color,

Every object has certain characteristics associated with it. The characteristics identify positions and contents. When an object is created Visual Basic automatically applies certain default properties. The user can change the properties of objects at design time or at run time.



**METHODS :**

These are not found in traditional programming. They represent statements or built in procedures that can be invoked to impart some action to a particular object. For example a method can be included to draw an object on the form; another method may print certain greetings on the form and so on. Different types of objects have different set of methods that can be used with them.

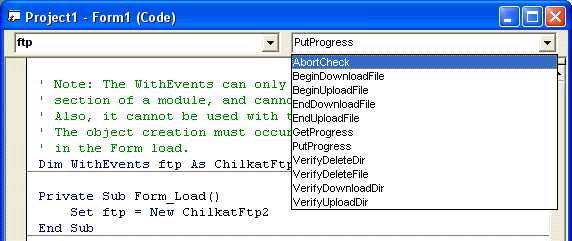
**PROCEDURES:**

All the code required for a particular application is written in either procedures or

subroutines. Code is normally related to some object or more accurately to the event of an object. The code is executed when a certain event occurs.

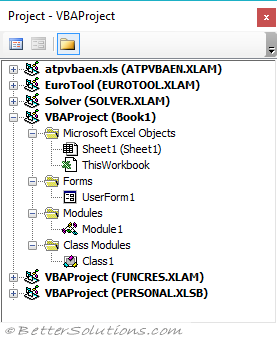
**MODULES :**

The code that is attached to a form is accessible from anywhere on that form. But a programs may be used in many forms. A module generally includes a collection of general procedures, variable declarations and constant definitions used by application. There may modules in one program and each module is saved as a separate file.



**PROJECT:**

The project holds together various forms and modules that make up a program. Its The purpose is primarily one of convenience. When you want to start work on a program you will have to open a project file. A project file has an extension of .VBP



**FORM WINDOW :**

The Form Window is central to developing Visual Basic applications. It is where you develop your application.

PROPERTIES :

Like all controls, the form has many properties. Fortunately, we only have to know about some of them. The properties we will be concerned with are:

Property (Description name) Name used to identify form. Three letter prefix for form Caption Text that appears in the title bar of form.

Icon Reference to icon that appears in title bar form.

Left form. Distance from left side of computer screen to left side of

Top form. Distance from top side of computer screen to top side of

Width of the form in tips

Height of form in tips.

Back Color Background color of form.

Border Style Form can either be sizable (can resize using the mouse) or fixed size.

Form Events

The form primarily acts as ‘container’ for other controls, but it does support events. That is, it can respond to some user interactions. We will only be concerned with two form events in this course:

Events Description

Click Event executed when user clicks on the form with the mouse.

Load Event executed when the form first loads the computer Memory. This is a good place initials values for various Properties and project values.

Toolbox Window:

The toolbox window is the selection menu for controls used in your application

**MICROSOFT ACCESS 2003:**

Microsoft Access is the default database of Microsoft Visual Basic, Microsoft Access 2003 provides many new features that make working with data and designing a database even easier. Microsoft Access Database is a collection of data and objects related to a particular topic or purpose. Microsoft Access Database may contain tables, queries, forms, reports, macros modules and shortcuts top data access pages. Microsoft Access is a Relational Database Management System. Using Access we can organize our data according to subject and can store information about how different subjects are related. In general MS-Access databases can have several small tables. Microsoft Office Access, previously known as Microsoft Access, is a relational database Management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. It is a member of the 2007 Access can use data stored in Access/Jet, Microsoft SQL Server, Oracle, or ODBC-compliant data container (including My SQL and Post grad SQL). Skilled software and data architects use it to develop application software. Relatively unskilled programmers and non-programmer “power users” can use it to build simple applications. It supports some object-oriented techniques but falls short of being a fully object-oriented development tool. Microsoft's first attempt to sell a relational database product was during the mid 1980s, when Microsoft obtained the license to sell. In the late 1980s Microsoft developed its own solution codenamed Omega. It was confirmed in 1988 that a database product for Windows and OS/2 was in development It was going to include the "EB" Embedded Basic language, which was going to the language for writing macros in all Microsoft applications but the unification of macro languages did not happen until the introduction of Visual Basic for Applications (VBA). Omega was also expected to provide a front end to the Microsoft SQL Server.. The application was very resource-hungry, and there were reports that it was working slowly on the processors that were available at the time .It was scheduled to be released in the 1st quarter of 1990, but in 1989 the development of the product was reset and it was rescheduled to be delivered no sooner than in January 1991 Parts of the project were later used for other Microsoft projects: Cirrus (codename for Access) and Thunder (codename for where the Embedded Basic engine was used).After Access's premiere, the Omega project was demonstrated in 1992 to several journalists and included features that were not available in Access. In addition to using its own database storage file, Microsoft Access also may be used as the 'front-end' of a program while other products actas the 'back-end' tables, such as Microsoft SQL Server and non-Microsoft products such as Oracle and Sybase.

Multiple backend sources can be used by a Microsoft Access Jet Database (ACCDB and MDB formats). Similarly, some applications such as Visual Basic, ASP.NET, or Visual Studio .NET will use the Microsoft Access database format for its tables and queries. Microsoft Access may also be part of a more complex solution, where it may be integrated with other technologies such as Microsoft Excel, Microsoft Outlook, Microsoft Word, Microsoft PowerPoint and ActiveX controls.

Access tables support a variety of standard field types, indices, and referential integrity including cascading updates and deletes. Access also includes a query interface, forms to display and enter data, and reports for printing. The underlying Jet database, which contains these objects, is multi-user and handles record-locking.

Repetitive tasks can be automated through macros with point-and-click options. It is also easy to place a database on a network and have multiple users share and update data without overwriting each other's work. Data is locked at the record level which is significantly different from Excel which locks the entire spreadsheet. There are template databases within the program and for download from Microsoft's website. These options are available upon starting Access and allow users to enhance a database with predefined tables, queries, forms, reports, and macros. Database templates support VBA code but Microsoft's templates do not include VBA code.

The number of simultaneous users that can be supported depends on the amount of data, the being performed, level of use, and Generally accepted limits are solutions with 1 GB or less of data (Access supports up to 2 GB) and it performs quite well with 100 or fewer simultaneous connections (255 concurrent users are supported). This capability is often a good fit for solutions. If using an Access database solution in a multi-user scenario, they should be "split". This means that the tables are in one file called the back end (typically stored on a shared network folder) and the application components (forms, reports, queries, code, macros, linked tables) are in another file called the front end. The linked tables in the front end point to the back end file. Each user of the Access application would then receive his or her own copy of the front end file.

Applications that run complex queries or analysis across large datasets would naturally require greater bandwidth and memory. Microsoft Access is designed to scale to support more data and users by linking to multiple Access databases or using a back-end database like Microsoft SQL Server. With the latter design, the amount of data and users can scale to enterprise-level solutions.

Microsoft Access's role in web development prior to version 2010 is limited. User interface features of Access, such as forms and reports, only work in Windows. In versions 2000 version2003 and Access object type called Data Access Pages created publishable web pages.

Data Access Pages are no longer supported. The Microsoft Jet Database Engine, core to Access, can be accessed through technologies such as ODBC or OLE DB. The data (i.e., tables and queries) can be accessed by web-based applications developed in ASP.NET, PHP, or Java. With the use of Microsoft's Terminal Services and Remote Desktop Application in Windows Server 2008 R2, organizations can host Access applications so they can be run over the web.[26] This technique does not scale the way a web application would but is appropriate for a limited number of users depending on the configuration of the host. Access 2010 allows databases to be published to SharePoint 2010 web sites running Access Services. These web-based forms and reports run in any modern web browser. The resulting web forms and reports, when accessed via a web browser, don't require any add-ins or extensions (e.g. ActiveX, Silverlight).

Access 2013 can create web applications directly in SharePoint 2013 sites running Access Services. Access 2013 web solutions store its data in an underlying SQL Server database which is much more scalable and robust than the Access 2010 version which used SharePoint lists to store data Access Services in SharePoint has since been retired.[27]A compiled version of an Access database (File extensions: .MDE /ACCDE or .ADE; ACCDE only works with Access 2007 or later) can be created to prevent user from accessing the design surfaces to modify module code, forms, and reports. An MDE or ADE file is a Microsoft Access database file with all modules compiled and all editable source code removed. Both the .MDE and .ADE versions of an Access database are used when end-user modifications are not allowed or when the application's source code should be kept confidential.

Microsoft also offers developer extensions for download to help distribute Access 2007 applications, create database templates, and integrate source code control with Microsoft Visual SourceSafe.

Microsoft also offers developer extensions for download to help distribute Access 2007 applications, create database templates, and integrate source code control with Microsoft Visual SourceSafe.

Users can create tables, queries, forms and reports, and connect them together with macros. Advanced users can use VBA to write rich solutions with advanced data manipulation and user control. Access also has report creation features that can work with any data source that Access can access.

The original concept of Access was for end users to be able to access data from any source. Other features include: the import and export of data to many formats including Excel, Outlook, ASCII, database, Paradox, FoxPro, SQL Server and Oracle. It also has the ability to link to data in its existing location and use it for viewing, querying, editing, and reporting. This allows the existing data to change while ensuring that Access uses the latest data.

It can perform heterogeneous joins between data sets stored across different platforms. Access is often used by people downloading data from enterprise level databases for manipulation, analysis, and reporting locally.

There is also the Jet Database format (MDB or ACCDB in Access 2007) which can contain the application and data in one file. This makes it very convenient to distribute the entire application to another user, who can run it in disconnected environments.

One of the benefits of Access from a programmer's perspective is its relative compatibility with SQL (structured query language)—queries can be viewed graphically or edited as SQL statements, and SQL statements can be used directly in Macros and VBA Modules to manipulate Access tables. Users can mix and use both VBA and "Macros" for programming forms and logic and offer object-oriented possibilities. VBA can also be included in queries.

Microsoft Access offers parameterized queries. These queries and Access tables can be referenced from other programs like VB6 and .NET through DAO or ADO. From Microsoft Access, VBA can reference parameterized stored procedures via ADO.

The desktop editions of Microsoft SQL Server can be used with Access as an alternative to the Jet Database Engine. This support started with MSDE (Microsoft SQL Server Desktop Engine), a scaled down version of Microsoft SQL Server 2000, and continues with the SQL Express versions of SQL Server 2005 and 2008.

Microsoft Access is a file server-based database. Unlike client–server relational database management systems (RDBMS), Microsoft Access does not implement database triggers, stored procedures, or transaction logging. Access 2010 includes table-level triggers and stored procedures built into the ACE data engine. Thus a Client-server database system is not a requirement for using stored procedures or table triggers with Access 2010. Tables, queries, forms, reports and macros can now be developed specifically for web based applications in access 2010. Integration with Microsoft SharePoint 2010 is also highly improved.

The 2013 edition of Microsoft Access introduced a mostly flat design and the ability to install apps from the Office Store, but it did not introduce new features. The theme was partially updated again for 2016, but no dark theme was created for Access.

Microsoft offers free runtime versions of Microsoft Access which allow users to run an Access desktop application without needing to purchase or install a retail version of Microsoft Access. This actually allows Access developers to create databases that can be freely distributed to an unlimited number of end-users. These runtime versions of Access 2007 and later can be downloaded for free from Microsoft [33] the runtime versions for Access 2003 and earlier were part of the Office Developer Extensions/Toolkit and required a separate purchase.

The runtime version allows users to view, edit and delete data, along with running queries, forms, reports, macros and VBA module code. The runtime version does not allow users to change the design of Microsoft Access tables, queries, forms, reports, macros or module code.

The runtime versions are similar to their corresponding full version of Access and usually compatible with earlier versions; for example Access Runtime 2010 allows a user to run an Access application made with the 2010 version as well as 2007 through 2000. Due to deprecated features in Access 2013, its runtime version is also unable to support those older features.

**3. SYSTEM DESIGN**

**Data Flow Diagram**

Customer

Login

Verification New customer added

Fitness Management System

New cust created

Verified

To

Transaction

Amount paid

Plan

Manages

Maintain the payment

Instructor

Add Plan Gives Instructions

**ER DIAGRAM FOR FITNESS MANAGEMENT SYSTEM**

have

Member

Trainer

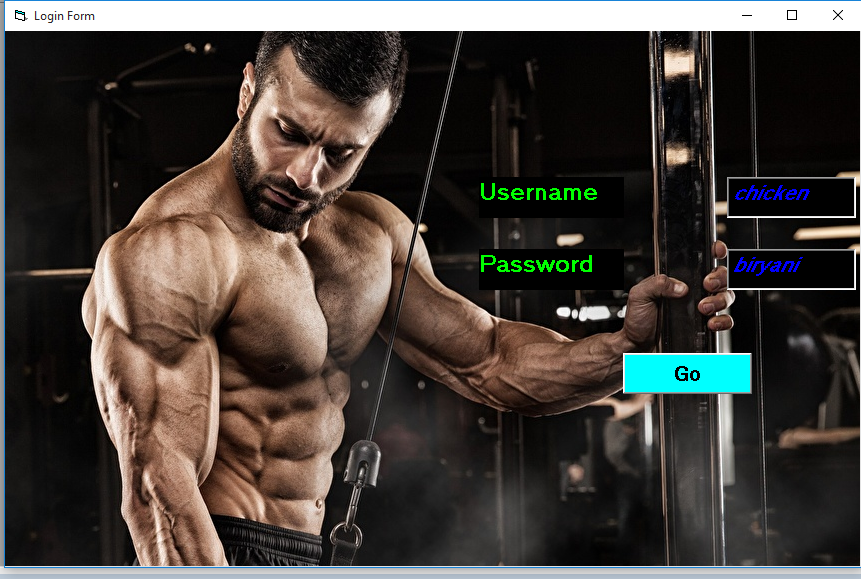
has

Plan

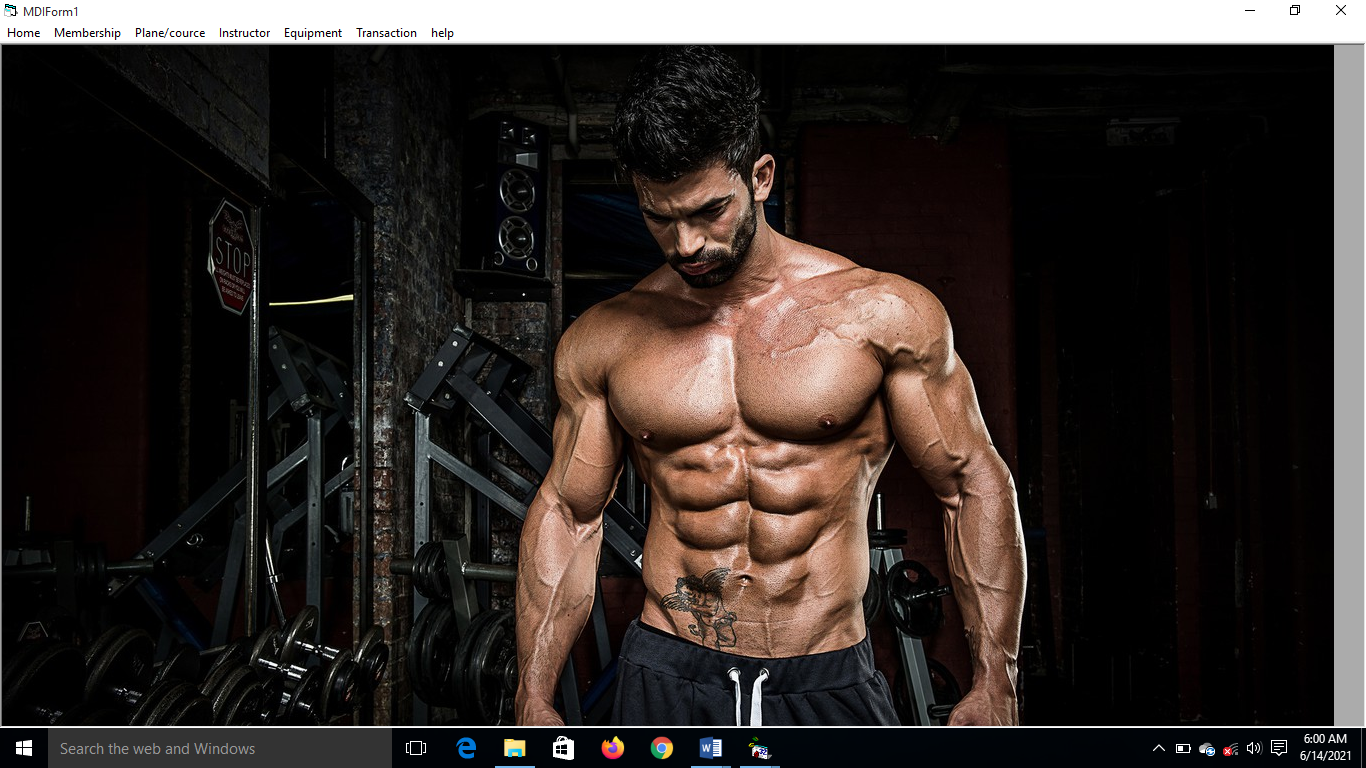
Equipment

Fitness center

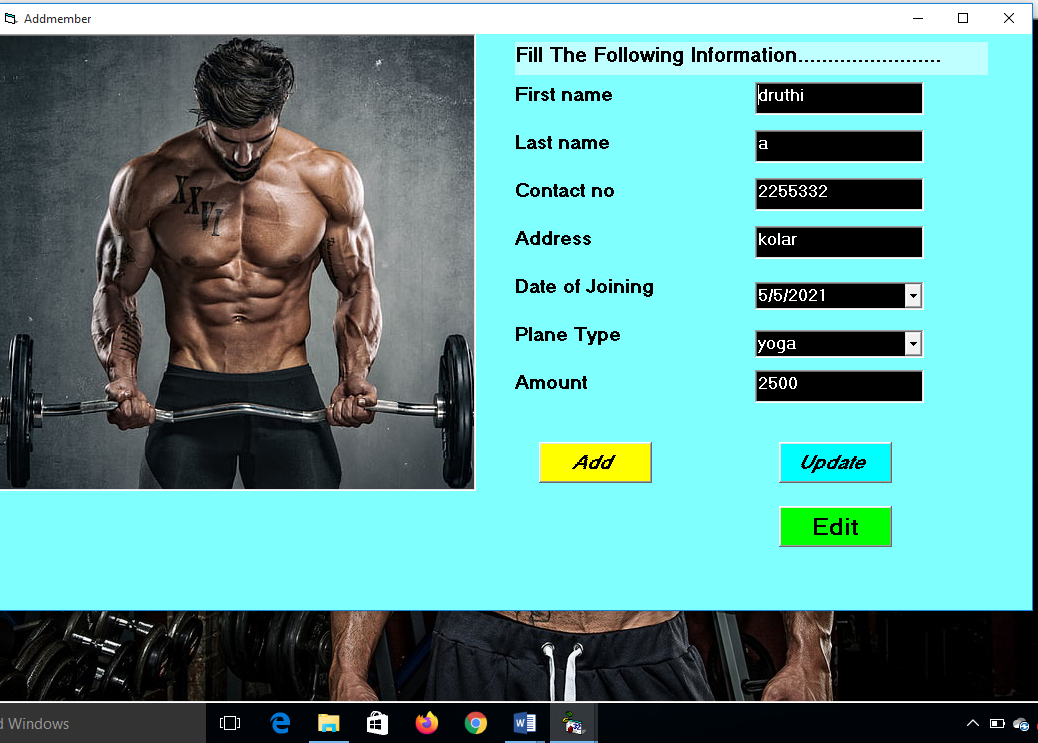
1. **Login Form :**



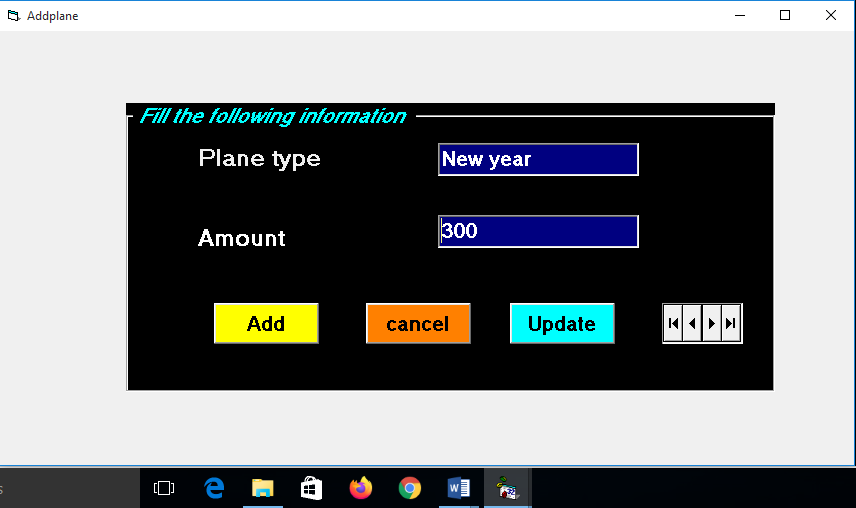
1. **Main Form :**

****

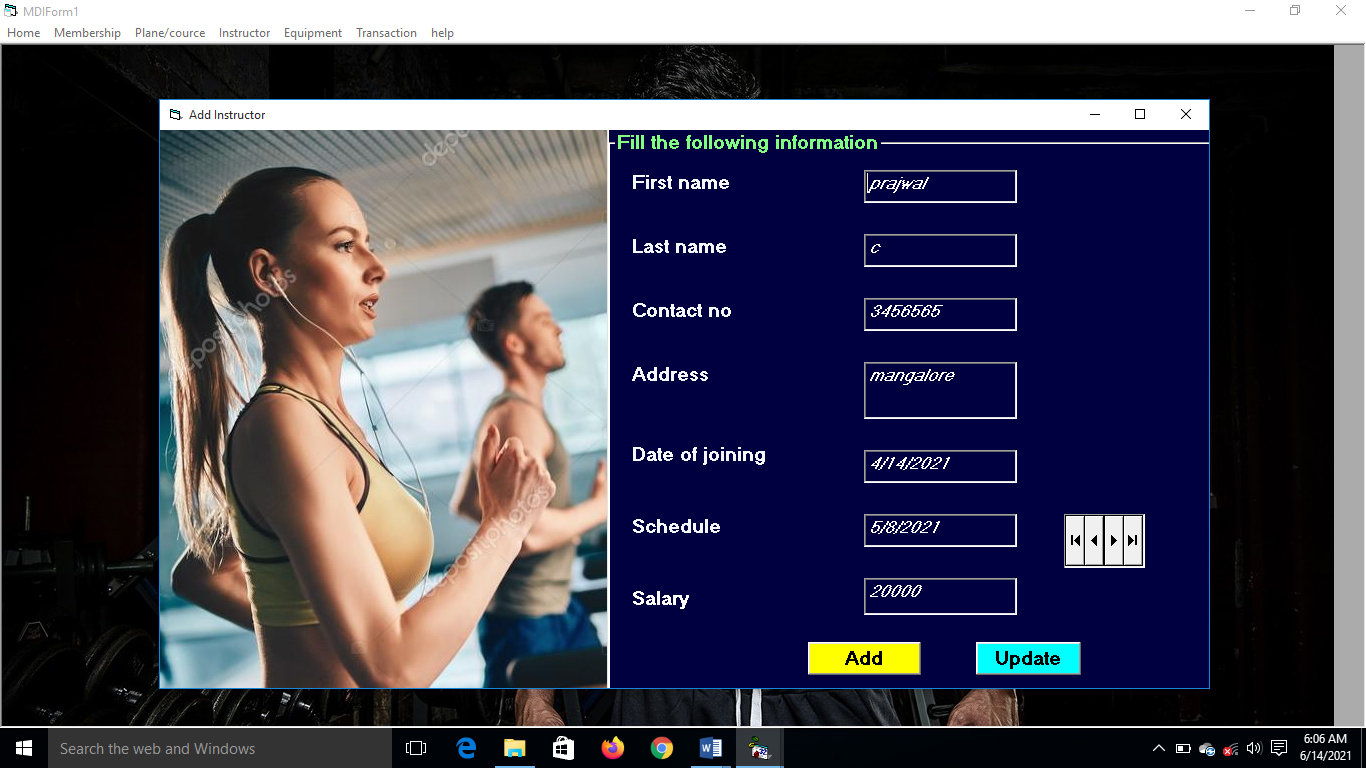
1. **Membership Form :**

****

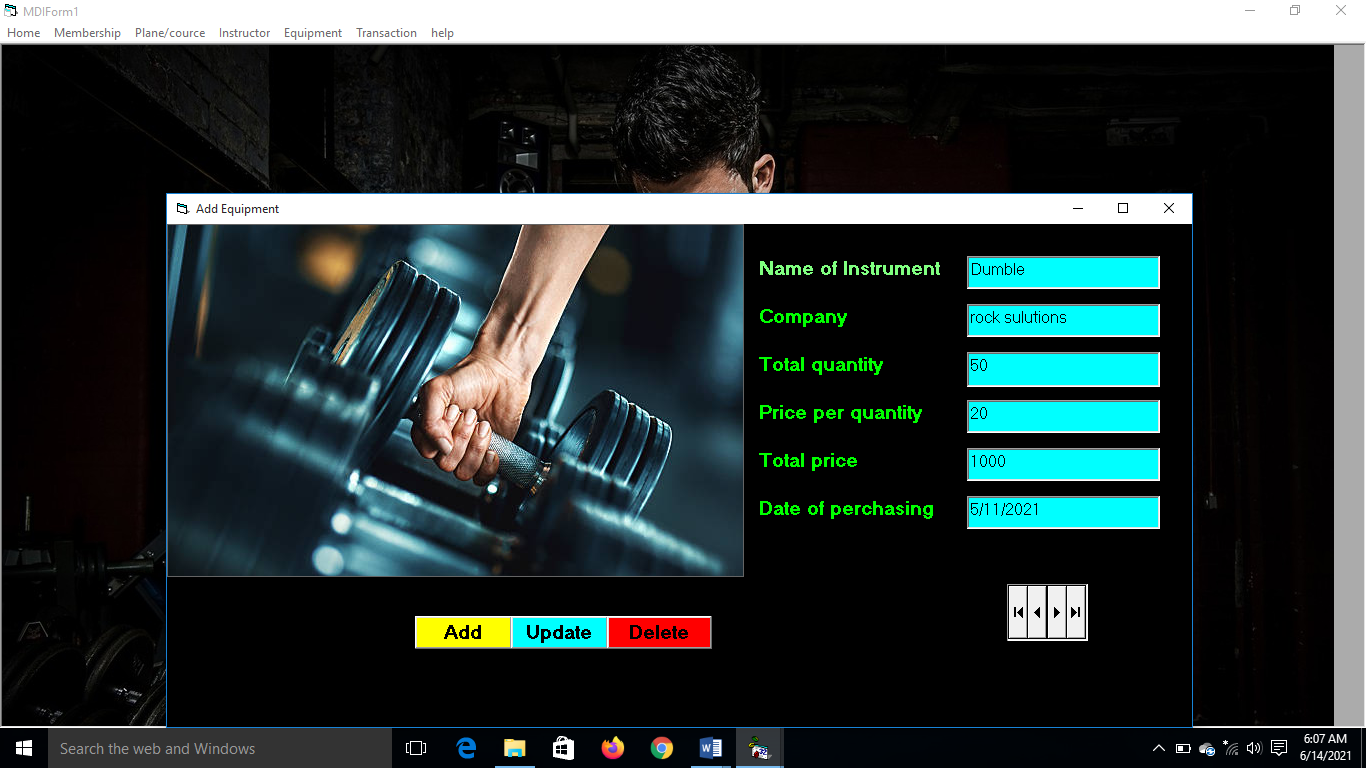
1. **Plan/cource Form :**

****

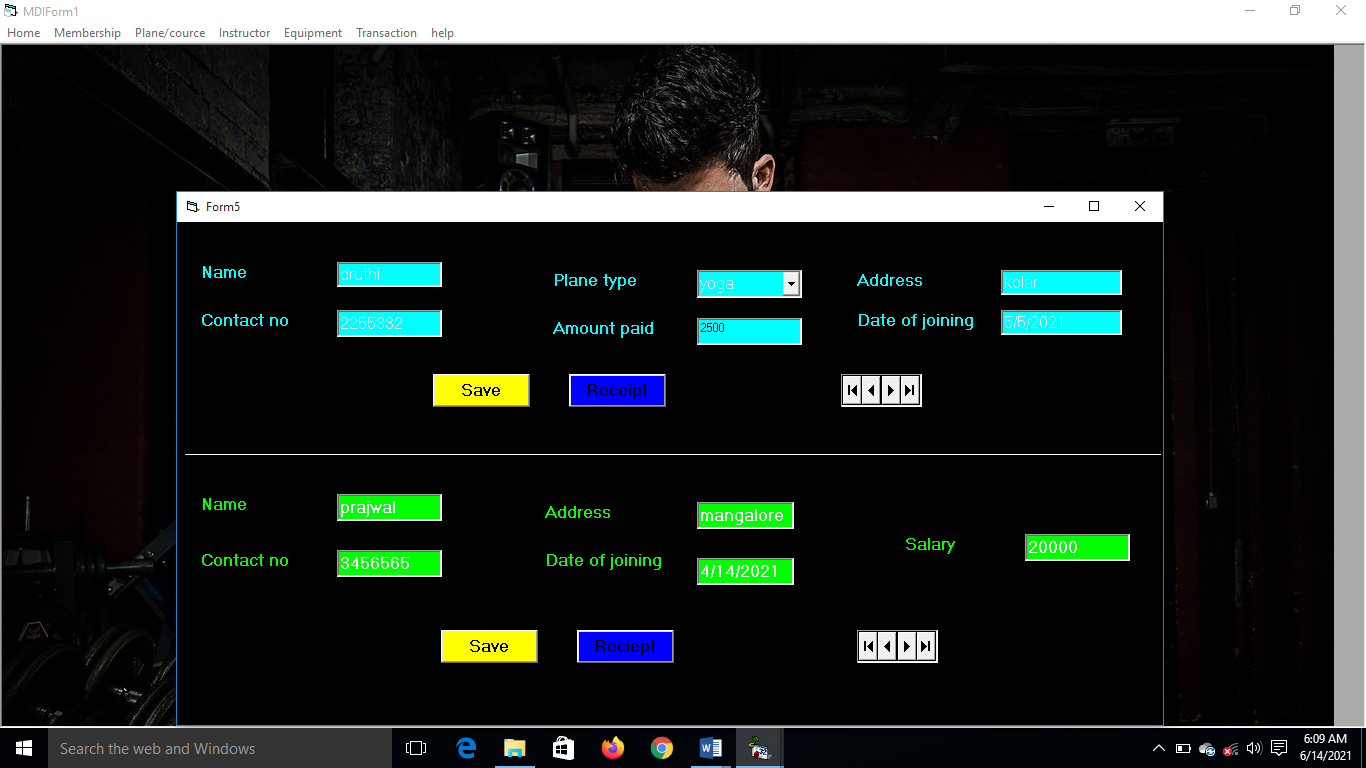
1. **Instructor Form :**

****

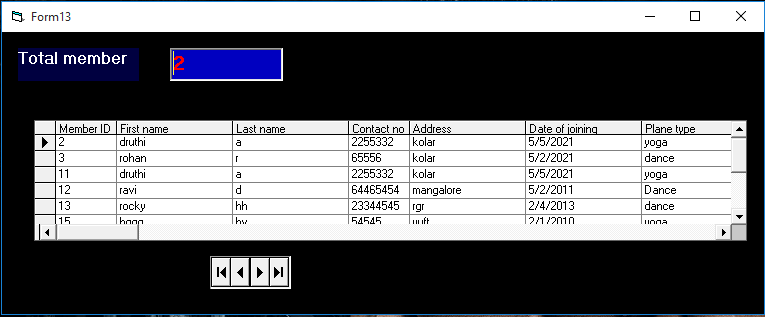
1. **Equipment Form :**

****

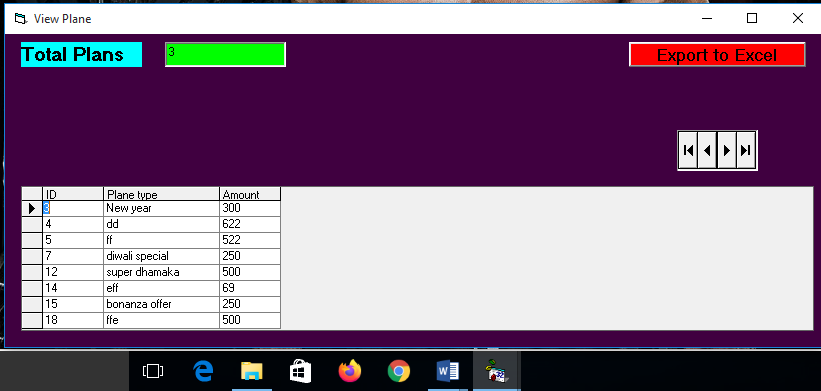
1. **Transaction Form :**

****

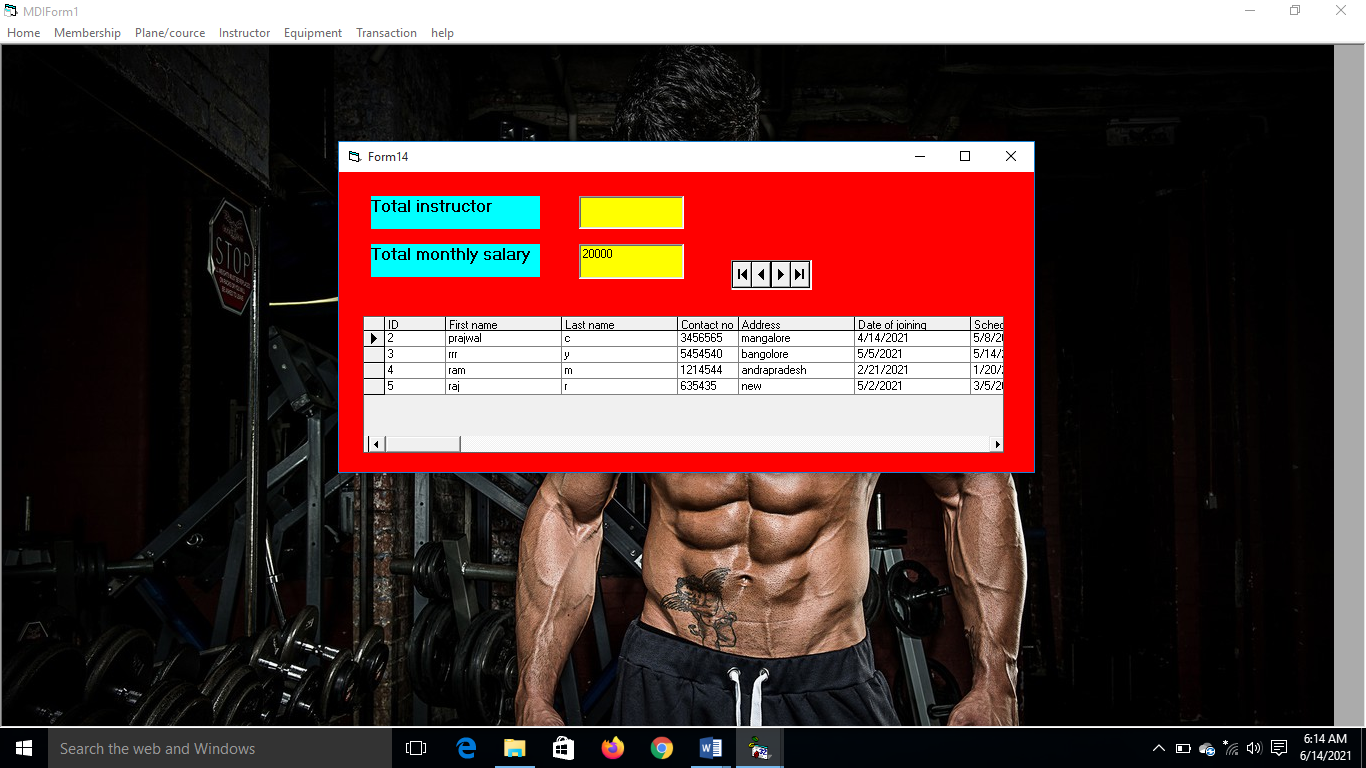
1. **Membership View :**

****

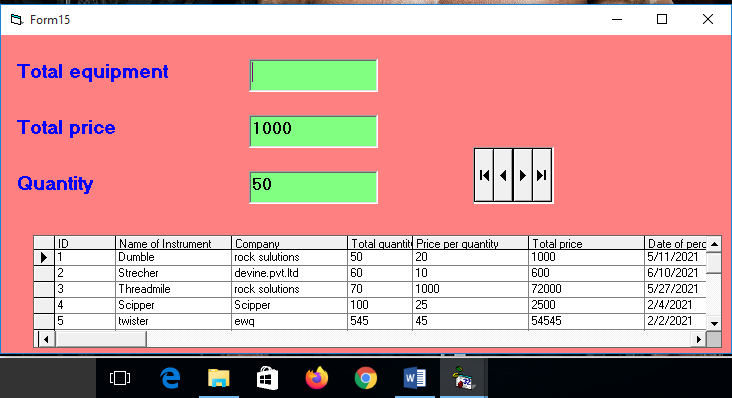
1. **Plane view :**



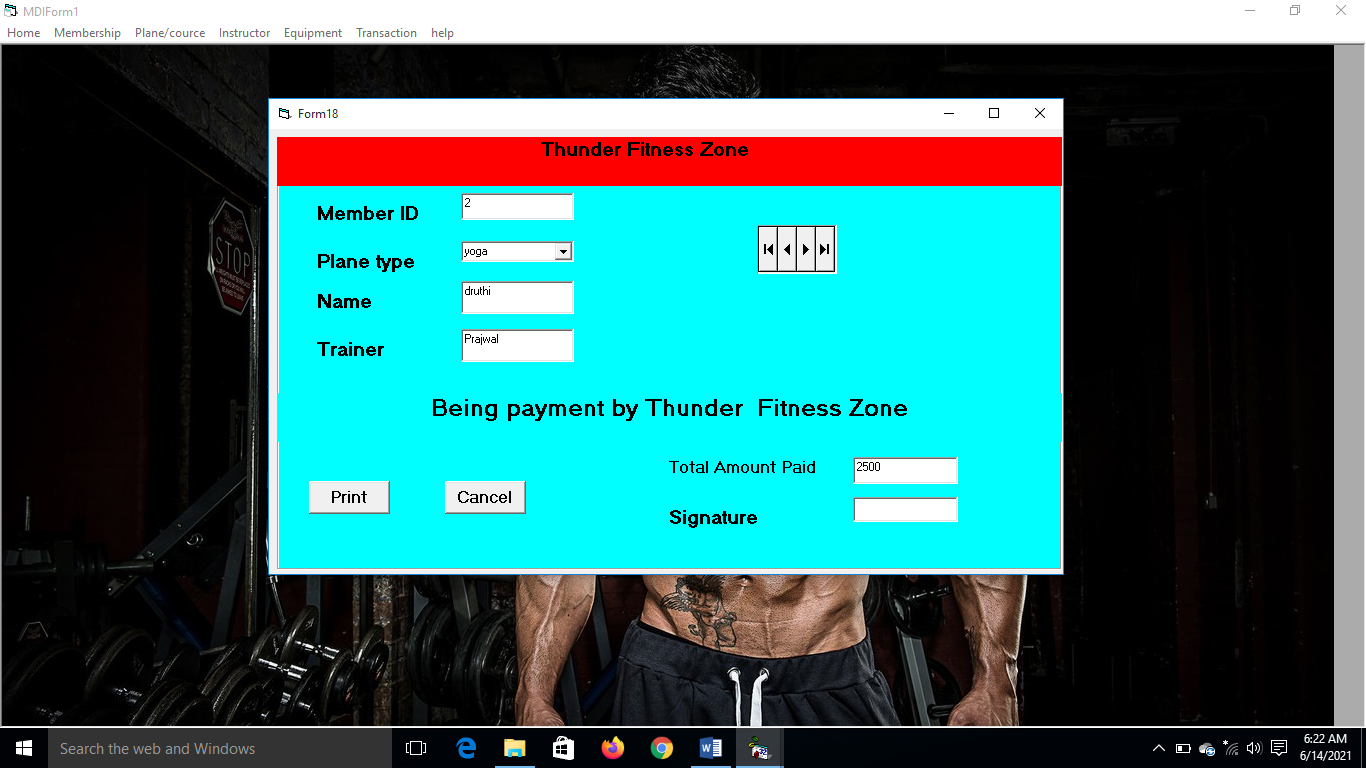
1. **Instructor View :**



1. **Equipment View :**



1. **Receipt Form :**



**Coding :**

1. **Login Form**

Private Sub Command1\_Click()

Dim username, password As String

username = "chicken"

password = "biryani"

If Text1.Text = username And Text2.Text = password Then

MsgBox ("sign in successfull")

MDIForm1.Show

Me.Hide

ElseIf Text1.Text <> username Or Text2.Text <> password Then

MsgBox ("sign in failes")

End If

End Sub

1. **Main form**

Private Sub e2\_Click()

Form9.Show

End Sub

Private Sub e3\_Click()

Form10.Show

End Sub

Private Sub e4\_Click()

Form15.Show

End Sub

Private Sub hh2\_Click()

Form12.Show

End Sub

Private Sub i2\_Click()

Form7.Show

End Sub

Private Sub i3\_Click()

Form8.Show

End Sub

Private Sub i4\_Click()

Form14.Show

End Sub

Private Sub m2\_Click()

form2.Show

End Sub

Private Sub m3\_Click()

Form3.Show

End Sub

Private Sub m4\_Click()

Form13.Show

End Sub

Private Sub p2\_Click()

Form4.Show

End Sub

Private Sub p3\_Click()

Form17.Show

End Sub

Private Sub p4\_Click()

Form6.Show

End Sub

Private Sub q1\_Click()

End

End Sub

Private Sub t2\_Click()

Form5.Show

End Sub

Private Sub t3\_Click()

Form16.Show

End Sub

Private Sub t4\_Click()

Form5.Show

End Sub

1. **Membership Form**

Private Sub Command1\_Click()

Adodc1.Recordset.Fields("First name") = Text1.Text

Adodc1.Recordset.Fields("Last name") = Text2.Text

Adodc1.Recordset.Fields("Contact no") = Text3.Text

Adodc1.Recordset.Fields("Address") = Text4.Text

Adodc1.Recordset.Fields("Date of joining") = Combo1.Text

Adodc1.Recordset.Fields("Plane type") = Combo2.Text

Adodc1.Recordset.Fields("Amount") = Text5.Text

Adodc1.Recordset.Update

MsgBox ("record sucessfully updates")

End Sub

Private Sub Command2\_Click()

Form3.Show

End Sub

Private Sub Command3\_Click()

Adodc1.Recordset.AddNew

End Sub

1. **Plane / cource Form**

Private Sub Command1\_Click()

Adodc2.Recordset.Fields("plane type") = Text1.Text

Adodc2.Recordset.Fields("Amount") = Text2.Text

Adodc2.Recordset.Update

MsgBox ("Offer successfully inserted")

End Sub

Private Sub Command3\_Click()

Adodc2.Recordset.AddNew

End Sub

Private Sub Command4\_Click()

confirmation = MsgBox("Do you want to delete this record", vbYesNo + vbCritical, "Delete record confirmation")

If confirmation = vbYes Then

Adodc2.Recordset.Delete

MsgBox "record has been deleted successfully", vbInformation, "message"

Else

MsgBox "record not deleted!!!", vbInformation, "message"

End If

Adodc2.Recordset.Delete

End Sub

1. **Instructor Form**

Private Sub addbtn\_Click()

Adodc3.Recordset.AddNew

End Sub

Private Sub Command3\_Click()

Adodc3.Recordset.Fields("First name") = Text1.Text

Adodc3.Recordset.Fields("Last name") = Text2.Text

Adodc3.Recordset.Fields("Contact no") = Text3.Text

Adodc3.Recordset.Fields("Address") = Text4.Text

Adodc3.Recordset.Fields("Date of joining") = Text5.Text

Adodc3.Recordset.Fields("Schedule") = Text6.Text

Adodc3.Recordset.Fields("Salary") = Text7.Text

Adodc3.Recordset.Update

MsgBox ("record successfully updated")

End Sub

1. **Equipment Form**

Private Sub Command2\_Click()

Adodc4.Recordset.Fields("Name of Instrument") = Text1.Text

Adodc4.Recordset.Fields("Company") = Text2.Text

Adodc4.Recordset.Fields("Total quantity") = Text3.Text

Adodc4.Recordset.Fields("Price per quantity") = Text4.Text

Adodc4.Recordset.Fields("Total price") = Text5.Text

Adodc4.Recordset.Fields("Date of perchasing") = Text6.Text

Adodc4.Recordset.Update

MsgBox ("record successfully updated")

End Sub

Private Sub Command3\_Click()

confirmation = MsgBox("Do you want to delete this record", vbYesNo + vbCritical, "Delete record confirmation")

If confirmation = vbYes Then

Adodc4.Recordset.Delete

MsgBox "record has been deleted successfully", vbInformation, "message"

Else

MsgBox "record not deleted!!!", vbInformation, "message"

End If

End Sub

1. **Transaction Form**

Private Sub Command1\_Click()

cc.Recordset.Fields("First name") = Text2.Text

cc.Recordset.Fields("Contact no") = Text3.Text

cc.Recordset.Fields("Address") = Text4.Text

cc.Recordset.Fields("Date of joining") = Text5.Text

cc.Recordset.Fields("Plane type") = Text6.Text

cc.Recordset.Fields("Amount") = Text7.Text

cc.Recordset.Update

MsgBox ("Member Fees has been saved")

End Sub

Private Sub Command2\_Click()

Form18.Show

End Sub

Private Sub Command4\_Click()

Form11.Show

End Sub

1. **Edit Membership**

Private Sub Command1\_Click()

Form3.Text2.Text = form2.Text1.Text

Form3.Text3.Text = form2.Text2.Text

Form3.Text4.Text = form2.Text3.Text

Form3.Text5.Text = form2.Text4.Text

Form3.Text6.Text = form2.Text5.Text

Form3.Combo1.Text = form2.Combo1.Text

Form3.Combo2.Text = form2.Combo2.Text

End Sub

Private Sub Command3\_Click()

Adodc6.RecordSource = "select \* from adodc1 where Member ID =" + Text7.Text + "or name=" '+text7.text+"'

Adodc6.Caption = Adodc6.RecordSource

End Sub

Private Sub Command4\_Click()

Adodc6.Recordset.AddNew

End Sub

Private Sub Command5\_Click()

Adodc6.Recordset.Fields("First name") = Text2.Text

Adodc6.Recordset.Fields("Last name") = Text3.Text

Adodc6.Recordset.Fields("contact no") = Text4.Text

Adodc6.Recordset.Fields("Address") = Text5.Text

Adodc6.Recordset.Fields("date of joining") = Combo1.Text

Adodc6.Recordset.Fields("plane type") = Combo2.Text

Adodc6.Recordset.Fields("Amount") = Text6.Text

Adodc6.Recordset.Update

MsgBox "Record successfully updated", vbInformation, "message"

End Sub

Private Sub Command6\_Click()

confirmation = MsgBox("Do you want to delete this record", vbYesNo + vbCritical, "Delete record confirmation")

If confirmation = vbYes Then

Adodc6.Recordset.Delete

MsgBox "record has been deleted successfully", vbInformation, "message"

Else

MsgBox "record not delete!!!", vbInformation, "message"

End If

End Sub

Private Sub Command7\_Click()

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text4.Text = ""

Text5.Text = ""

Combo1.Text = ""

Combo2.Text = ""

End Sub

1. **Edit Plane**

Private Sub Frame1\_DragDrop(Source As Control, X As Single, Y As Single)

End Sub

Private Sub Command1\_Click()

gg1.Recordset.Fields("plane type") = Text1.Text

gg1.Recordset.Fields("Amount") = Text2.Text

gg1.Recordset.Update

MsgBox ("Offer successfully inserted")

End Sub

Private Sub Command2\_Click()

confirmation = MsgBox("Do you want to delete this record", vbYesNo + vbCritical, "Delete record confirmation")

If confirmation = vbYes Then

gg1.Recordset.Delete

MsgBox "record has been deleted successfully", vbInformation, "message"

Else

MsgBox "record not delete!!!", vbInformation, "message"

End If

End Sub

1. **Edit Instructor**

Private Sub Command1\_Click()

instruct1.Recordset.Fields("First name") = Text2.Text

instruct1.Recordset.Fields("Last name") = Text3.Text

instruct1.Recordset.Fields("Contact no") = Text4.Text

instruct1.Recordset.Fields("Address") = Text5.Text

instruct1.Recordset.Fields("Date of joining") = Combo1.Text

instruct1.Recordset.Fields("Schedule") = Text7.Text

instruct1.Recordset.Fields("Salary") = Text8.Text

instruct1.Recordset.Update

MsgBox ("record successfully updated")

End Sub

Private Sub Command2\_Click()

confirmation = MsgBox("Do you want to delete this record", vbYesNo + vbCritical, "Delete record confirmation")

If confirmation = vbYes Then

instruct1.Recordset.Delete

MsgBox "record has been deleted successfully", vbInformation, "message"

Else

MsgBox "record not delete!!!", vbInformation, "message"

End If

End Sub

Private Sub Command3\_Click()

instruct1.Recordset.AddNew

End Sub

Private Sub Command4\_Click()

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Combo1.Text = ""

Text7.Text = ""

Text8.Text = ""

End Sub

1. **Update Equipment**

Private Sub Command1\_Click()

instrument1.Recordset.Fields("Name of Instrument") = Text2.Text

instrument1.Recordset.Fields("Company") = Text3.Text

instrument1.Recordset.Fields("Total quantity") = Text4.Text

instrument1.Recordset.Fields("Price per quantity") = Text5.Text

instrument1.Recordset.Fields("Total price") = Text6.Text

instrument1.Recordset.Fields("Date of perchasing") = Text7.Text

instrument1.Recordset.Update

MsgBox ("record successfully updated")

End Sub

Private Sub Command2\_Click()

confirmation = MsgBox("Do you want to delete this record", vbYesNo + vbCritical, "Delete record confirmation")

If confirmation = vbYes Then

instrument1.Recordset.Delete

MsgBox "record has been deleted successfully", vbInformation, "message"

Else

MsgBox "record not delete!!!", vbInformation, "message"

End If

End Sub

Private Sub Command3\_Click()

instrument1.Recordset.AddNew

End Sub

Private Sub Command4\_Click()

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

End Sub

4. SYSTEM TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrast to the actual requirements.

Applications of Software Testing

* Cost Effective Development - Early testing saves both time and cost in many aspects,

however reducing the cost without testing may result in improper design of a software

application rendering the product useless.

* Product Improvement - During the SDLC phases, testing is never a time-consuming

process. However diagnosing and fixing the errors identified during proper testing is

a time-consuming but productive activity.

* Test Automation - Test Automation reduces the testing time, but it is not possible to

start test automation at any time during software development. Test automaton

should

be started when the software has been manually tested and is stable to some extent.

Moreover, test automation can never be used if requirements keep changing.

* Quality Check - Software testing helps in determining following set of properties of any software such also Functionality

o Reliability

o Usability

o Efficiency

o Maintainability

o Portability

Testing performs a very critical role for quality assurance and ensuring the reliability of the software. The success of testing for errors in program depends critically on the test cases.

The basic levels of testing are:

1. Unit testing

2. Integrating testing

3. System testing

4. Acceptance testing

**SYSTEM TESTING**

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting anomaly for the software. The testing phase involves testing of a system using various test data. Preparation The test data plays a vital role in the system testing. After preparation the test data, the system

Under study is tested using those test data Errors were found and corrected by using the following testing steps and corrections are recorded for future reference. Thus, a series of testing is performed on the system before it is ready for implementation.

The development of software system involves a series of production activities where opportunities for injection of human errors are enormous. Errors may begin to occur at the very inception of the process where the objectives may be enormously or imperfectly specified as well in later design and development stages. Because of human inability to perform and communicate with perfection, software development is followed by assurance activity.

Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. And of course it includes assurances that the system meets the specification and the requirements for the intended use and performance. The various levels of quality assurance are described in the following sections.

**TEST CASE DESIGN**

The test case design for software should be done in such a way that the errors can be found out with a minimum amount of time and effort. A test case is a set of data that the system will process as normal as input. Mainly there are two methods for the test case design-Black box tests and White box Testing.

**BLACK BOX TESTING**

This method focuses on the functional requirements of the software. It attempts to find

out the errors of the following categories-incorrect and missing functions, interface errors, errors in data structures or external database access, performance errors and initialization and termination errors.

**WHITE BOX TESTING**

The development of software system involves a series of production activities where opportunities for injection of human errors are enormous. Errors may begin to occur at the very inception of the process where the objectives may be enormously or imperfectly specified as well in later design and development stages. Because of human inability to perform and communicate with perfection, software development is followed by assurance activity Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. And of course it includes assurances that the system meets the specification and the requirements for the intended use and performance. The various levels of quality assurance are described in the following sections.

**INTEGRATION TESTING**

Data can be lost across an interface; one module can have an adverse effect on another’s sub functions, when combined may not produce the desired major function; global data structures can present problems. Integration testing is a symmetric technique for

Constructing tests to uncover errors associated with the interface. All modules are combined in this testing step. Then the entire program was tested as a whole.

**VALIDATION TESTING**

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors were uncovered and corrected and a final series of software validation began. Here we test the system in a manner that can be reasonably expected by the customer, the system was tested against system requirements.

**OUTPUT TESTING**

After performing validation test the next phase in output test of the system, since no system could be useful if it does not produce the desired output in a desired format. By considering the format of the report/output, output/ report is generated or displayed and is tested. Here output format is considered in two ways: one is the screen and other is a printed form.

**TESTING PROCESS**

First thing in the testing process is to make plans. In the test plan, test cases are selected to ensure that there ii\s an error in the program then it is executed by one of the test cases. The success of testing depends critically on these test cases selected.

**TEST CASE SPECIFICATION**

It is the major activity in the testing process. Test cases are specified for testing each unit, the specification gives all the test cases inputs to be used and the outputs expected for these test cases.

**5.Coclution and future enhance**

Using Visual basic and MS access we successfully completed all the modules and all the modules are working properly we are checked with giving dummy data, did not trace any errors

Since it is the first version of the software the performance of the software was measured and we found that all the modules are working fine.

In future the same modules will be expanded with online facilities as of now transaction details are maintained. In future through the software only we will give a members to pay their training amount and varies training modules will be introduced in online mode.

**BIBOLOGRAPHY**

Visual Basic

Lewis, Thomas. VB COM. Chicogo, IL; Wrox press., 1999

Pattinson, ted. Programming Distributed Applications with COM and Microsoft Visual Basic 6.0. Redmond, WA: Microsoft Press, 1998.

Atamatakis, Williams. Microsoft visual Basic Design Partterns. Redmond, WA: Microsoft press, 2000.

Net references

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

[www.ask.com/Reference+Books](http://www.ask.com/Reference+Books)

[www.visualbasic.about.com](http://www.visualbasic.about.com)

[www.booksnbytes.com/authors](http://www.booksnbytes.com/authors)