MCS-044 (Mini Project)

(MCA)

Cricket Training Management System

Submitted by

Enrollment No-196636560
Name-Anikita Singhal
Contact No-8447705651
Email id - ani.singhal178@gmail.com
RC Code-29:Delhi 2(Rajghat)
Study centre code- Tecnia institute 29010

Abstract

Cricket Training Management System is used to improve the quality of training as per our curriculums. There are many teams and each of them need different training, different set of exercises, and different diet. With your system it should be possible to select a set of exercises and create a programme for each team according to their age and experience, and keep track of each team member and his/her performance. Also, it has the attendance system to record, who did not turn up for a particular session. This system is also helpful to prepare a diet chart for each and every member considering his or her age, height, weight, role, level etc.

It consists different types of modules to maintain the activities of software such as member addition, updating, team update and addition, diet chart preparation, attendance status add, performance rating add etc.

Acknowledgements

I would like to express my greatest appreciation to the all teachers who have helped and supported me throughout the project. I am thankful to my computer teacher for his ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project

A special acknowledgement goes to my classmates who helped me in completing the project by exchanging interesting ideas and sharing their experience.

Synopsis

Title

Cricket Training Management System

[1] INTRODUCTION

1.1 BACKGROUND:

We are going to Design and develop a Cricket Training Management System to improve the quality of training as per our curriculums. There are many teams and each of them need different training, different set of exercises, and different diet. With your system it should be possible to select a set of exercises and create a programme for each team according to their age and experience, and keep track of each team member and his/her performance. Also, it has the attendance system to record, who did not turn up for a particular session. This system is also helpful to prepare a diet chart for each and every member considering his or her age, height, weight, role, level etc.

It consists different types of modules to maintain the activities of software such as member addition, updating, team update and addition, diet chart preparation, attendance status add, performance rating add etc.

1.2 AIM & OBJECTIVE

- To reduce manual work.
- To achieve the goal of university
- All data are available in a place.
- To avoid Mistakes.
- This software requires less manpower and low maintains.
- Information can retrieve any time according to requirement.
- To minimization of managing problem.

1.3 PURPOSE AND SCOPE

System is designed that keeps track of team, team members, diet chart for team and members using this automated system. Analyse the system requirements, and design the system and submitted for academic evaluation.

2. Survey of Technology

2.1 TOOLS / PLATFORM, LANGUAGES

The Microsoft .NET Framework

The .NET Framework is the infrastructure for the Microsoft .NET platform. The .NET Framework is an environment for building, deploying, and running Web applications and Web Services. Microsoft's first server technology ASP (Active Server Pages) was a powerful and flexible "programming language". But this was too code oriented. This was not an application framework and not an enterprise development tool.

The Microsoft .NET Framework was developed to solve this problem.

.NET Frameworks keywords:

- Easier and quicker programming
- Reduced amount of code
- Declarative programming model
- Richer server control hierarchy with events
- Larger class library
- Better support for development tools

Windows 10

This has overcome the limitation of the previous operating systems. The 32-bits and 64 bits concept ensure that the applications working on this platform are more stable and hence have very little chances of 'Hanging'. It is multi-tasking and multiprocessing operating system to perform different task simultaneously.

C#.Net

This is a new programming language that is used for developing program for the Microsoft .NET. This is object oriented programming language. We can design web

application by using C-sharp. This is similar to c++ except that this supports HTML documents.

SQL Server

This is backend where the data is stored here and is really much secured database management system. This supports Client/Server concept also. most of the company prefers secured database management system than to cost. If can be used for distributed programming as well. This supper very easy query language like SQL the most popular query language. Now this supports Object Oriented Database and Knowledge Base Database Management System.

Ado.Net

This new data component, introduced with .NET, presented an exciting new approach to data access. Though the techniques, and logic used to connect to databases with ADO.NET weren't startlingly different from those used with its predecessor, ADO.NET had a lot to offer. What was unique about this technology was the architecture beneath This all, its powerful approach to data management, and the flexibility in the next level of data-presenting devices.

2.2 REASON FOR USING C# AS FRONT-END*:-*

.Net Framework is of the most prevailed framework to develop the desktop-based application. This is the outcome of Microsoft which was developed for competing java in the world market. It supports many programming languages like C++, C#, Visual Basic, XML etc.

- It is Very easy to use ASP.net to develop web-based application because of its user-friendly functionalities.
- C# is the language that uses both CUI and GUI Interfaces thus more flexibility.
- Auto-generated and More Powerful IDE.

- Common to use Server; every language uses Internet Information Server

2.3 REASON TO OPT SQL SERVER AS BACK-END: -

SQL Server is RDBMS tool which has been used by me as back-end due to following reason:

- In today's competitive environment, an organization wants a comprehensive, secure, reliable, and productive data platform for its business applications. SQL Server provides all these facilities.
- SQL Server 2012 combines data analysis, reporting, integration, and notification services.
- The SQL Server database Engine provides a platform that allows managing data application very easily

3. Requirement and Analysis:

3.1 PROBLEM DEFINITION:

Design and develop a Cricket Training Management System to improve the quality of training. Assume there are many teams (according to their age and experience) and each of them need different training, different set of exercises, and different diet. With your system it should be possible to select a set of exercises and create a programme for each team according to their age and experience, and keep track of each team member and his/her performance. Also, it should include the attendance system to record, who did not turn up for a particular session. Your system should also prepare a diet chart for each and every member considering his or her age, height, weight, role, level etc

3.2 REQUIREMENT SPECIFICATION:

It consists different types of modules to maintain the activities of software such as member addition, updating, team update and addition, diet chart preparation, attendance status add, performance rating add etc.

3.3 PLANNING AND SCHEDULING:

- We prepare plan using following steps: -
- Prepare all requirements of organization on a paper.
- Identify the related tasks.
- Acquiring and organizing the tools and resources for the project.
- Preparation of well-defined schedule for events of the project
- Proper evaluation of progress of project development.
- Establishing various standards for the project by which we can find the standard output.

Work-Break Down Structure: - this is the scheduling technique where the project is scheduled in various phase following the top-down or bottom-up approach. The tree like structure is followed without any loop. At each phase or step, milestones and deliverables are mentioned with respect to requirements. The work break-down structure shows the overall breakup flow of the project and does not indicate any parallel flow.

3.4 REQUIREMENTS (H / S)

HARDWARE

Processor	: -	I-3
Clock speed	: -	3.0 GHz
HDD	: -	320 GB
CD ROM	: -	52 X
RAM	: -	1 GB
Monitor	: -	15'' Color

SOFTWARE

Platform	: -	Dot Net
Operating system	: -	Windows 10
Front-end	:-	C#.net 2012
Middle-end	:-	Ado.Net
Back-end	:-	SQL Server

TECHNICAL REQUIREMENT

- Productivity : - Proposed software must accomplish the all tasks of Institution according to the management.
- Reliability :- Operation must performs complete or accurate task.
- Security :- all information must be secure.
- Scalability :- the architecture adopted to develop this project must have technical flexibility to adopt the change in future.
- Integration : - all modules must be property integrated with each-other so that performance of system must be good.

3.5 PROCESS FLOW

1. **Interface Module:** it contains a list of all modules. It is designed with good interface architecture so that anyone can understand it easily. User can easily select the module and perform respective task.
2. **Member Module:** User can add member details including team details. User can update the member.
3. **Team module:** it is used to add/ update details of team.
4. **Training programme module:** it is used to add programme details as team. Team is selected from team data table.
5. **Attendance Module:** It is used to add attendance status and performance rating of members.
6. **Diet plan module:** it is used to add diet plan details, update and delete unused diet plan as per requirements.

3.6 INPUT INTERFACE

- Main Menu contains with various Options
- Add new Record
- Search Record
- Delete Record
- The system will be having user privileges-based menu.
- User will have to select the options form the given menu.
- The system will be entering the information into the database to generate reports.
- The forms will be designed to enter the data.
- Buttons will be used to insert, retrieve or modify the data.
- Links will be provided to shift from one form to another.

3.7 List of reports

- **Training programme**

  This is used to display training programme details such as programme id, date time and team.

- **Diet chart plan**

  It displays diet chart plan as per member and team.

3.8 DATA STRUCTURE

Team

Attributes	Datatype	Size	Constraints	description
Team_id	Int		Primary Key	This is team id number
Team_name	Varchar	50	Not Null	Team name
Age_group	Varchar	50	Not Null	Team for which Age group
Descr	Varchar	200		Description
Diet_plan_id	Int		Foreign key(diet_plan)	Diet_plan_id selected for team.
Experience	Varchar	100	Not Null	Experience details

Team_Member

Attributes	Datatype	Size	Constraints	Description
Mem_id	Int		Primary key	Member id
Team_id	Int		Foreign key (Team)	Team id number
mname	Varchar	50	Not Null	Member name
Email	Varchar	50	Not Null	Email ID
Address	Varchar	100	Not Null	Address
Contact	Varchar	10	Not Null	Contact number
Age	Int		Not Null	Age of member
Height	Int		Not Null	Height of member
Weight	Int		Not Null	Weight of member
Role	Varchar	50		Roles in team

Diet_plan

Attributes	Datatype	Size	Constraints	Description
Diet_plan__id	Int		Primary key	Diet plan ID
Breakfast	Varchar	200	Not Null	Breakfast diet
Lunch	Varchar	200	Not Null	Lunch diet details
Eve_meal	Varchar	200	Not Null	Evening meal
Post_training	Varchar	200	Not Null	Post training meal

Training_programme

Attributes	Datatype	Size	Constraints	Description
Prg_id	Int		Primary key	Programme ID
Team_id	Int		Foreign key (team)	Team id
tdate	Date		Not Null	Training date
Time	Date/Time		Not Null	Timing
Exercise_details	Varchar	200	Not Null	Exercise details
			Not Null	

Attendance_performance

Attributes	Datatype	Size	Constraints	Description
Attd_id	Int		Primary key	Attendance id
Prg_id	Int		Foreign key (training programme)	Programme id
Mem_id	Int		Foreign key	Member ID
status	Varchar		Not Null	Attend status(present/Absent)
Performance	Varchar		Not Null	Performance details (Excellent/ Very good/good/poor)

3.9 DFD

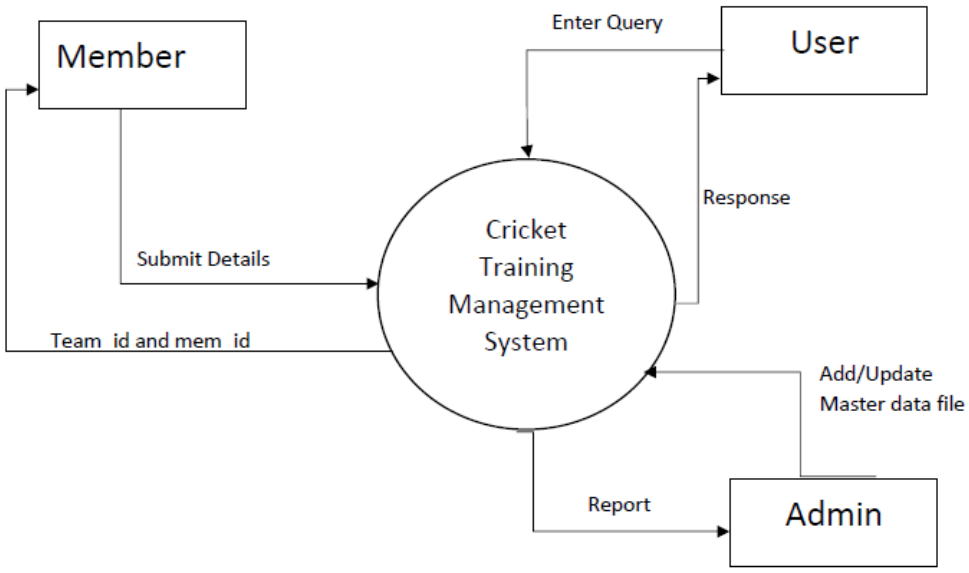
Data Flow

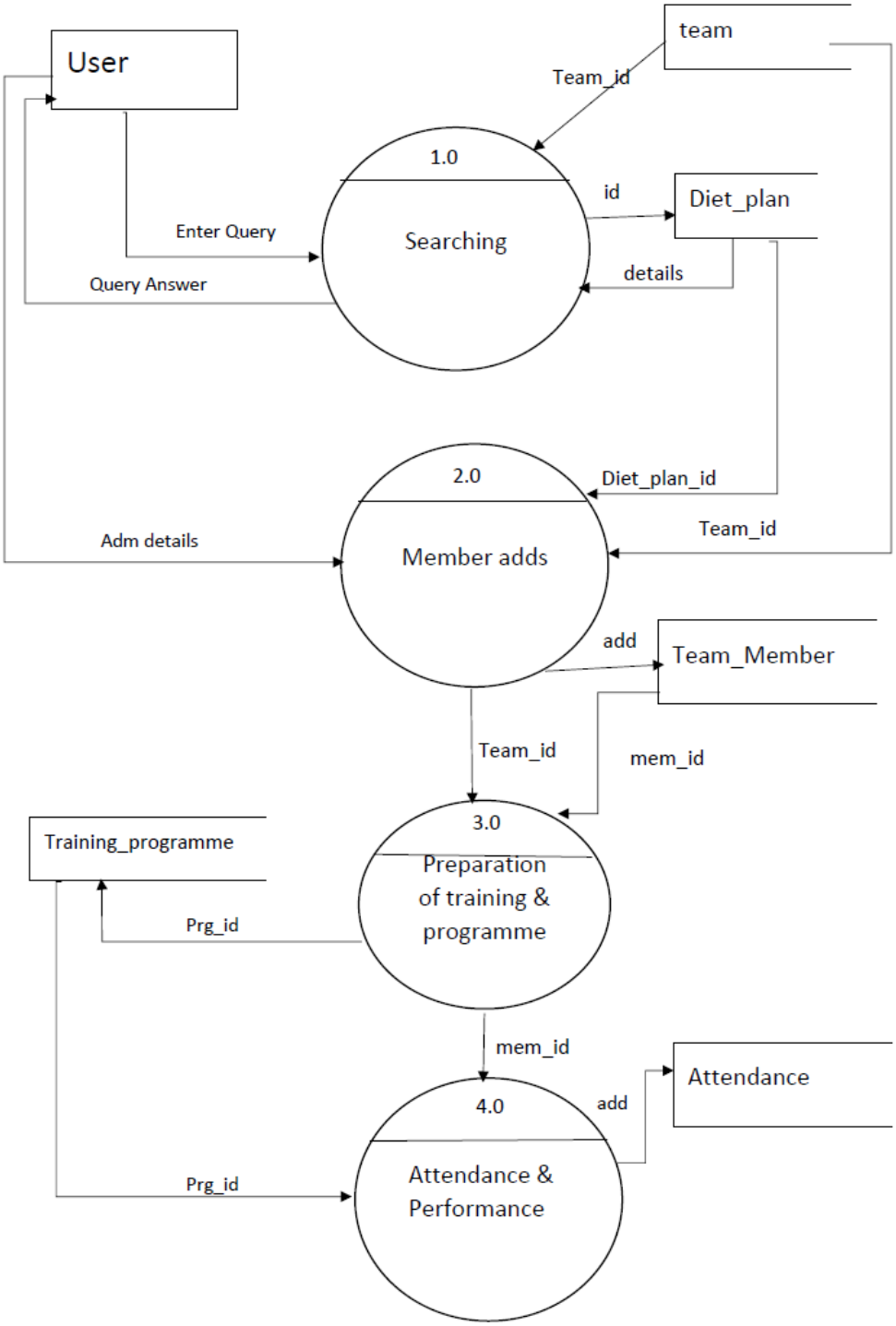
An arrow represents data flow; it represents the path over which data travels in the system. A data flow can move between processes, flow into or out of data stores, to and from external entities.

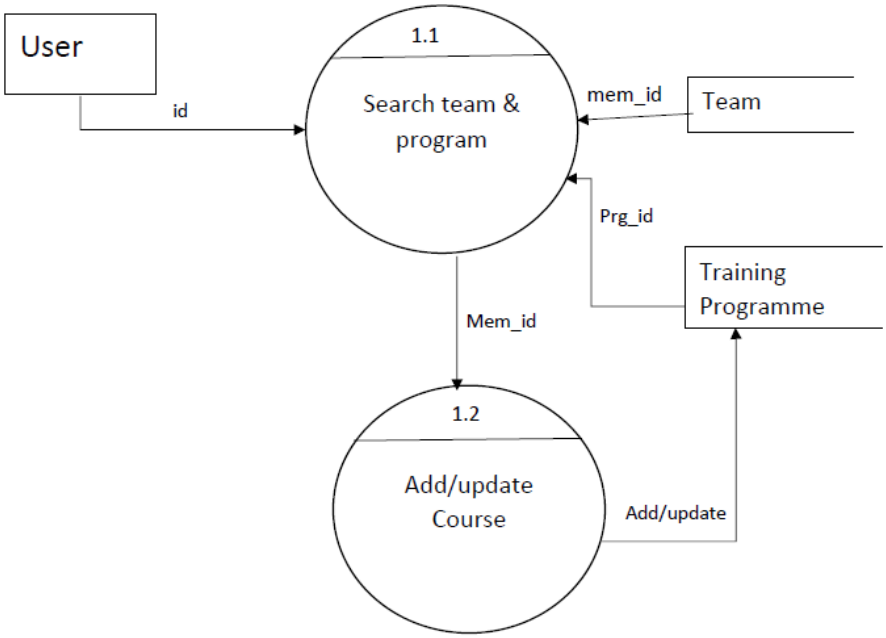
0 LEVEL DFD: -

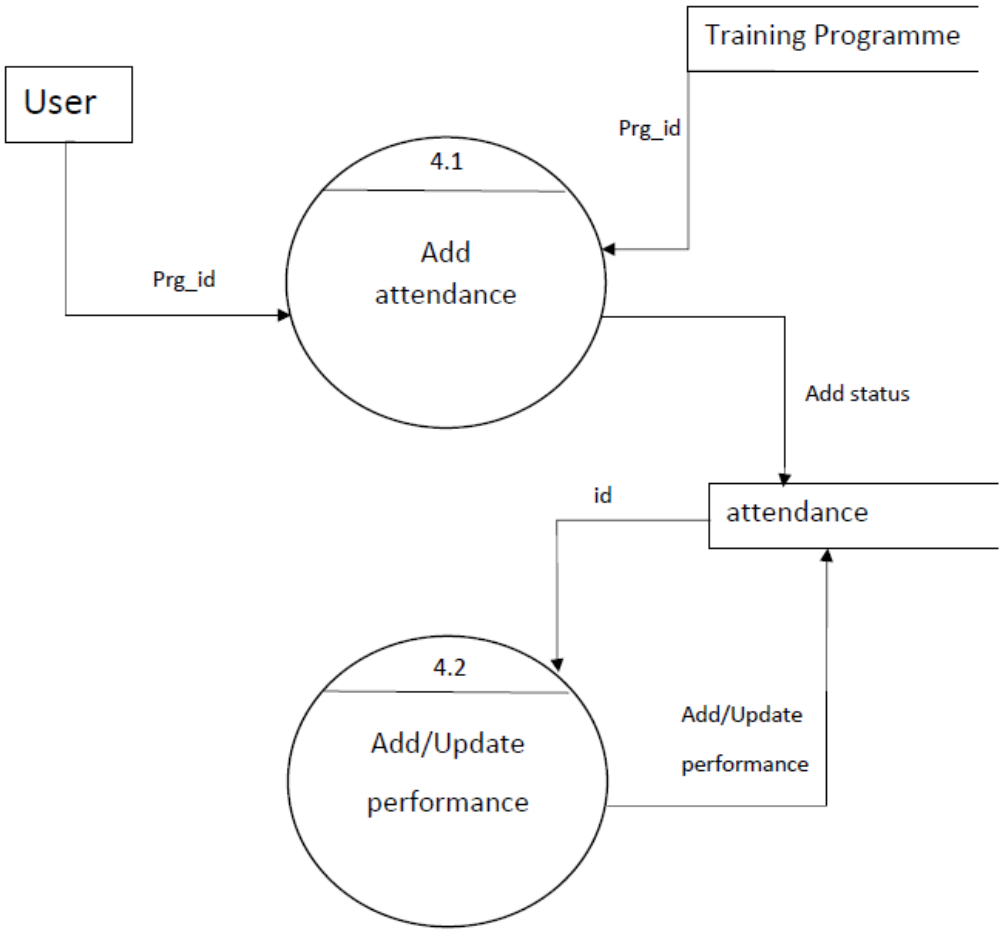
This is the context level DFD of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The purposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system

DFD level 0:

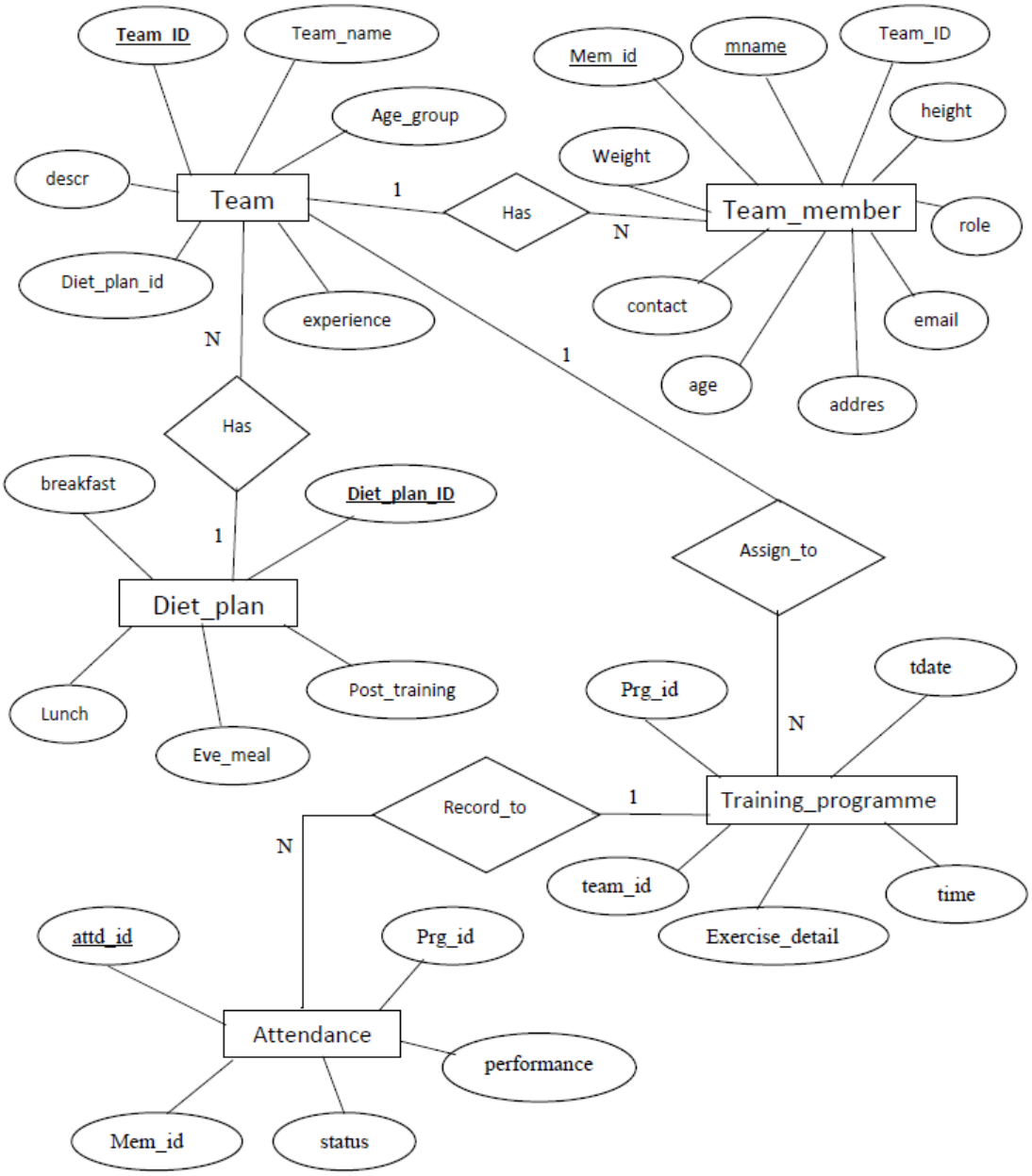



User
team
Team_id
1.0
Searching
id
Diet_plan
Enter Query
Query Answer
details
2.0
Member adds
Diet_plan_id
Team_id
Adm details
add
Team_Member
Team_id
mem_id
3.0
Preparation of training & programme
Training_programme
Prg_id
mem_id
4.0
Attendance & Performance
add
Attendance
Prg_id

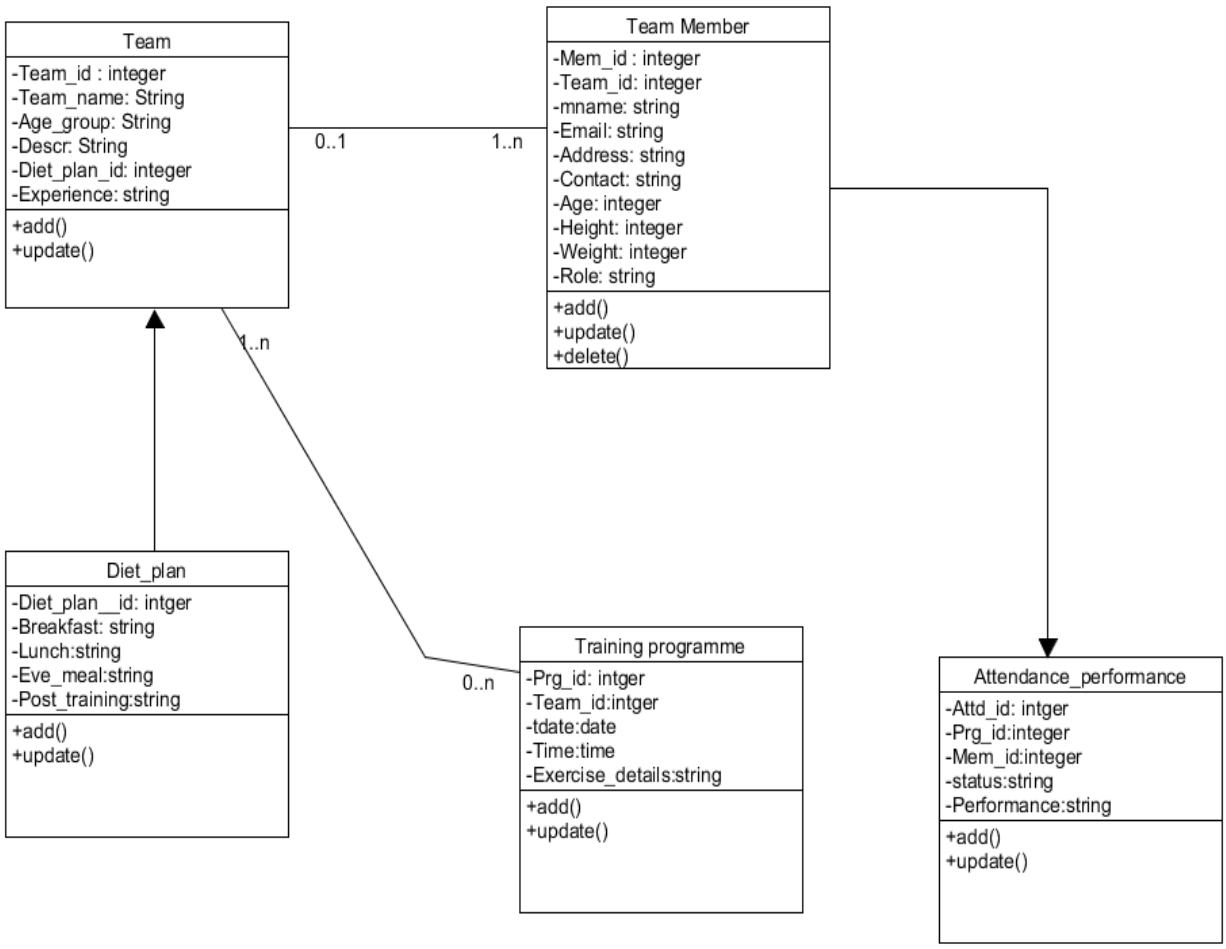

User
id
1.1
Search team &
program
mem_id
Team
Prg_id
Training
Programme
Mem_id
1.2
Add/update
Course
Add/update


Training Programme
User
Prg_id
4.1
Add
attendance
Prg_id
Add status
id
attendance
4.2
Add/Update
performance
Add/Update
performance

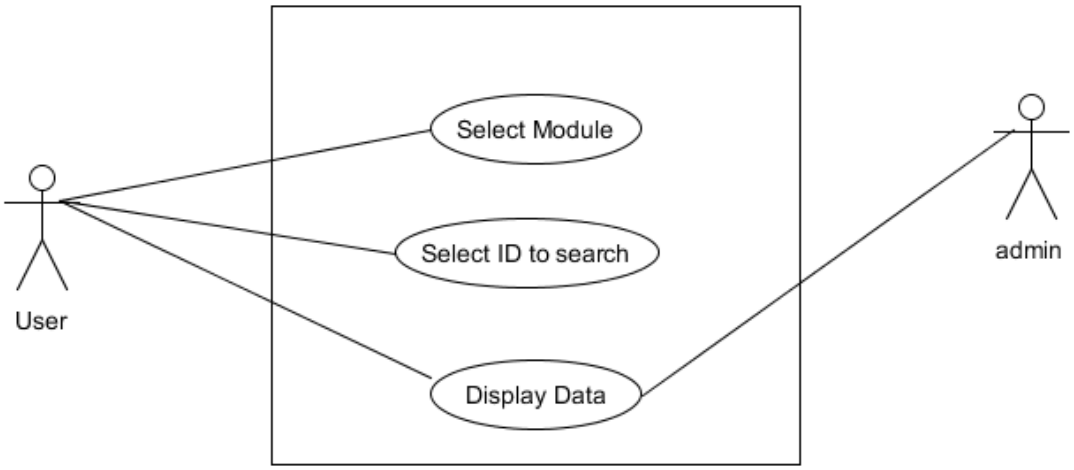
3.10 ER Diagram

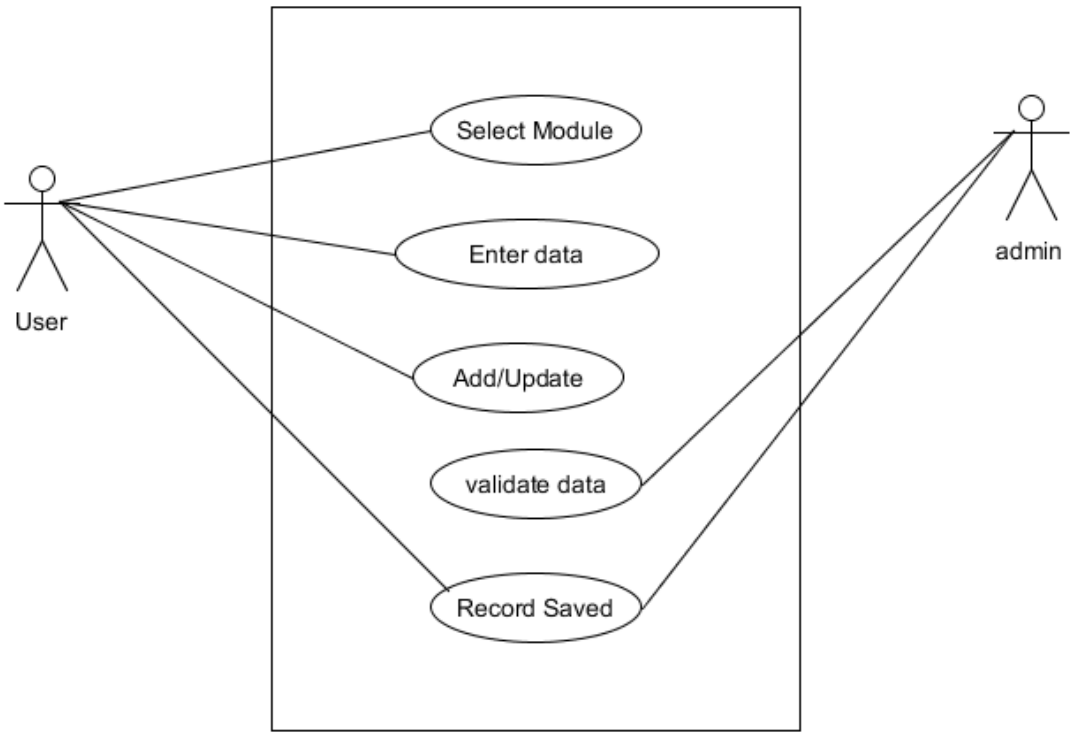


Class Diagram



Use Case


Select Module
Select ID to search
Display Data
User
admin


Select Module
Enter data
Add/Update
validate data
Record Saved
User
admin

3.11 FUTURE SCOPE & Limitation OF APPLICATION

The project entitled "**Cricket Training Management System**" meets all requirements of cricket training management. proposed software application is used to improve the training quality of cricket team. It uses large database where large amount of data is stored. Currently it is not able to manage other management activities and trainer details. It can be added as per requirements.

4. BIBLIOGRAPHY (References)

1. Software Engineering, A Practitioner's Approach by **Roger S. Pressman**
2. An Introduction to Database Systems by **Bipin C Desai**
3. Professional C# 2nd edition by **Wrox.**
4. **Microsoft SQL server by NIIT**.

Project Report

CHAPTER -1: INTRODUCTION

1.1Background:

We are going to Design and develop a Cricket Training Management System to improve the quality of training as per our curriculums. . There are many teams and each of them need different training, different set of exercises, and different diet. With your system it should be possible to select a set of exercises and create a programme for each team according to their age and experience, and keep track of each team member and his/her performance. Also, it has the attendance system to record, who did not turn up for a particular session. This system is also helpful to prepare a diet chart for each and every member considering his or her age, height, weight, role, level etc.

It consists different types of modules to maintain the activities of software such as member addition, updating, team update and addition, diet chart preparation, attendance status add, performance rating add etc.

1.2 OBJECTIVES

Main Summarized Objectives are: -

- To reduce manual work.
- To achieve the goal of organization.
- To reduce delay of customer service.
- All data are available in a place.
- To improve security level.
- To avoid mistakes.
- This software requires less manpower and low maintains.
- Information can retrieve any time according to requirement.
- To minimization of managing problem.

1.3 Purpose, Scope, and Applicability

The proposed project System meets all requirements of management system. It uses large database where large amount of data is stored.

An evaluation is carried out from the computer professional points of view. Design of the system and quality of the programming have to with stand the rigorous of careful

assessment, programming standard is quite high today and a modular approach. Internal and external training in efficient program writing technique can achieve surprisingly good result. The organizations requirements change after certain interval of time and the system is required to be copyright.

1.3.1 Purpose:

Proposed software is designed to reduce manual works and maintains the task accurately and timeliness.

1.3.2 Scope:

The project entitled "**Cricket Training Management System**" meets all requirements of training management process. It uses large database where large amount of data is stored.

The organizations requirements change after certain interval of time and the system is required to be copyright. This software is suitable to solve the future problem. It does not require any special Hardware requirement, hence can be run on any common architect of computer.

1.3.3 Applicability:

This system needs a database server on which the system is being run. The .Net platform is used in this project.

1.4 Achievement

The user of the system achieves more accuracy in work performance. The usage of computer in routine works will lead to accuracy in work; humans can do mistakes whereas previously tested software programs in an accurate way will never do mistakes.

1.5 Organisation of Report

This report describes the function and structure behavior of proposed software. It is developed for the activities of assignement submission.

.

CHAPTER-2: SURVEY OF TECHNOLOGY

2. SURVEY OF TECHNOLOGIES:

The Microsoft .NET Framework

The .NET Framework is the infrastructure for the Microsoft .NET platform. The .NET Framework is an environment for building, deploying, and running Web applications and Web Services. Microsoft's first server technology ASP (Active Server Pages) was a powerful and flexible "programming language". But it was too code oriented. It was not an application framework and not an enterprise development tool.

The Microsoft .NET Framework was developed to solve this problem.

.NET Frameworks keywords:

- Easier and quicker programming
- Reduced amount of code
- Declarative programming model
- Richer server control hierarchy with events
- Larger class library
- Better support for development tools

Windows 10

Windows 10 is an operating system produced by Microsoft for use on personal computers, including home and business desktops, laptops, net books, tablet PCs, and media centre PCs. Windows 8's new features are advances in touch and handwriting recognition support for virtual hard disks improved performance on multi-core processors, improved boot performance, Direct Access, and improvements. Windows 10adds support for systems using multiple heterogeneous graphics cards from different vendors (Heterogeneous Multi-adapter), a new version of Windows Media Centre& a Gadget for Windows Media Centre, improved media features.

C#.Net

It is a new programming language that is used for developing program for the Microsoft .NET. it is object-oriented programming language. We can design web application by using C-sharp. It is similar to c++ except that it supports HTML documents.

SQL Server 2008

It is backend where the data is stored here and is really much secured database management system. It is developed by Microsoft. It supports Client/Server concept also. Most of the company prefers secured database management system than to cost. If can be used for distributed programming as well. It supports very easy query language like SQL the most popular query language. Now it supports Object Oriented Database and Knowledge Base Database Management System.

Reason for using C# as Front-End:-

.Net Framework is of the most prevailed framework to develop the desktop-based application. This is the outcome of Microsoft which was developed for competing java in the world market. It supports many programming languages like C++, C#, Visual Basic, XML etc.

- It is Very easy to use ASP.net to develop web-based application because of its user-friendly functionalities.
- C# is the language that uses both CUI and GUI Interfaces thus more flexibility.
- Auto-generated and More Powerful IDE.
- Common to use Server; every language uses Internet Information Server

Reason to Opt SqlServer 2008 as Back-End:-

Sql Server 2008 is RDBMS tool which has been used by me as back-end due to following reason:

- In today's competitive environment, an organization wants a comprehensive, secure, reliable, and productive data platform for its business applications. Sql Server 2008 provides all these facilities.
- It combines data analysis, reporting, integration, and notification services.
- The Sql Server 2008 database Engine provides a platform that allows managing data application very easily.

Chapter-3

REQUIREMENTS & ANALYSIS

System analysis is the first step towards the software building process. The purpose of system analysis is to understand the system requirements, identify the data, functional and behavioral requirements and building the models of the system for better understanding of the system.

3.1 Problem Definition

In the process of system analysis, one should first understand that, what the present system, what it does, is how it works (i.e. processes). After analyzing these points, we become able to identify the problems the present system is facing. Upon evaluating current problems and desired information (input and output to the system), the analyst looks towards one or more solutions. To begin with, the data objects, processing functions, and behavior of the system are defined in detail. After this, a model from three different aspects of the system data, function and behavior the models created during the system analysis process helps in better understanding of data and control flow, functional processing, operational behavioral and information content.

3.2 Requirement Specification: -

It is quite difficult and time-consuming task to find the information as well as maintaining information manually. If all these information are to be kept at a single place it is also not possible in the manual system. Computerized system will upgrade and manage information very easily.

Software required for the system is also different from a normal desktop system. First of all a server software will be mandatory (here Internet Information Server (IIS)).

3.3 Planning and Scheduling

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure.

Structure of project plan: -

- **Introduction: -** This section states the objective of the project and lists the constraints such as budget and time.
- **Project Estimation: -** It states the details of estimation techniques to be used.
- **Project resources: -** states people, hardware and software
- **Risk management Strategy: -** It describes the possible project risks.
- **Project Schedule: -** It states the breakdown structure of the project.

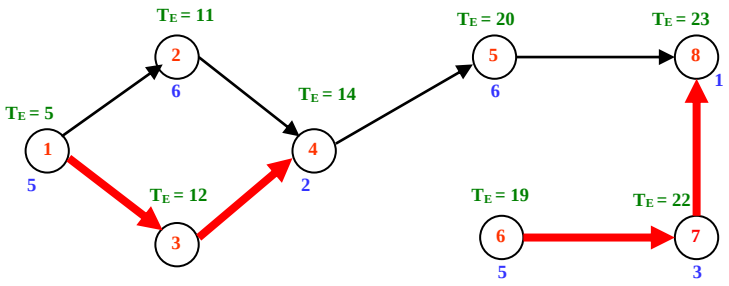
- **Project / staff organization:** - It describes the structure of development team and management reporting.

Project planning takes following steps: -

- Organizing the resources available for the project.
- scheduling the events of the project
- evaluating the process
- Establishing standard for the project

Project Planning & Scheduling

- PERT (Program Evaluation and Review Technique) CHART. PERT" developed by the United States Department of Defense as a management tool for complex military projects is an acronym for "Program Evaluation and Review Technique".
- Pert charts are used for project scheduling
- Pert charts allow software planners, or individuals to:
- Determine the critical path a project must follow.
- Establish most likely time estimates for individual task by applying statistical models.
- Calculate boundary times that define a time ‘window’ for a particular task



The critical path represents the shortest time, in which a project can be completed. Any activity on the critical path that is delayed in completion delays the entire project. Activities not on the critical path contain slack time and allow the project manager some flexibility in scheduling.

Gantt chart

A visual representation of a project over time

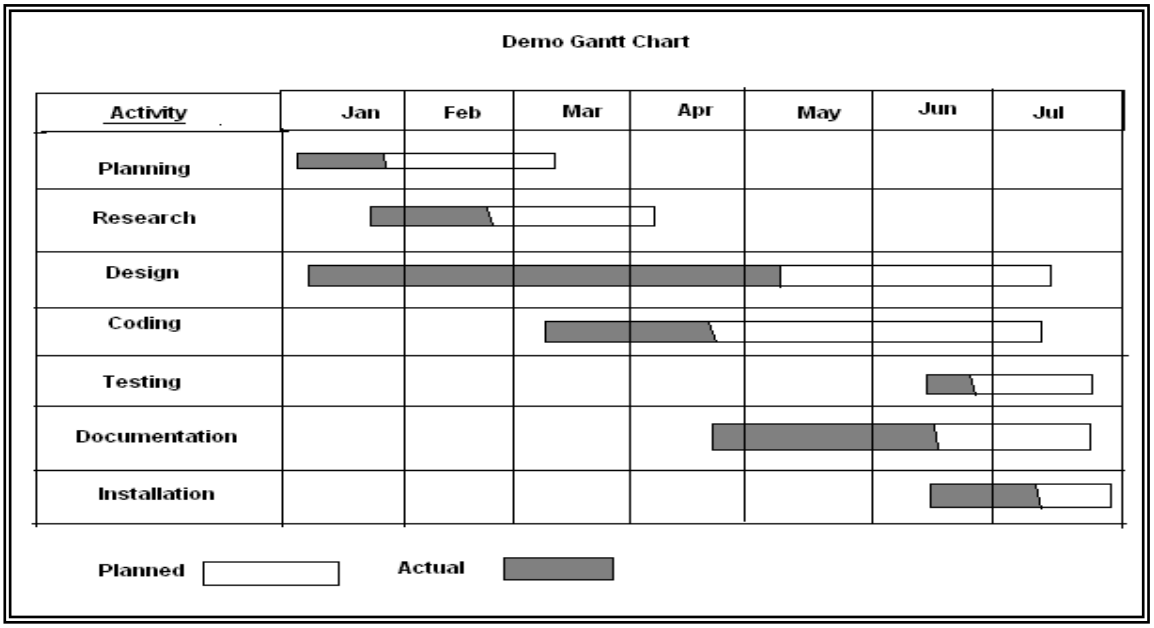
- Used to schedule, coordinate, and allocate the resources needed to complete a project.
- Shows start, end, and specific target dates called “milestones”.

Benefits of Gantt Charts

- Efficiency Increase
- Project tracking
- Set deadlines
- Communication Increase
- Coordination Increase
- Provides motivation through scheduling
- Encourages creativity

Disadvantages of Gantt Charts

- Activity descriptions often lack detail
- A lack of precedent and subsequent task relationships
- Does not allow for uncertain situations such as late or early finish times.



3.4 Software and Hardware Requirements

HARDWARE

Processor	: -	I-3
Clock speed	: -	3.0 GHz
HDD	: -	180 GB
CD ROM	: -	52 X
RAM	: -	4GB
Monitor	: -	15'' color
Other	: -	Printer, scanner, keyboard, mouse,

SOFTWARE

Platform	: -	Windows, .NET
Operating system	: -	Windows 10.
Font-end	:-	C#.net 2012
Back-end	:-	SQL Server2012

3.5 Preliminary product Decription: - System Analysis is not only time consuming but also a rigorous task. But it is crucial and most important phase of Software development process. Preliminary Investigation is the process of

gathering dataor requirement analysis. It is more helpful for problem definition and requirement specification.

Feasibility Study

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. This type of study determines if a project can and should be taken. Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements.

Different Type of Feasibility Study: -

In the conduct of the feasibility study, the analyst will usually consider seven distinct, but inter- related types of feasibility. They are

- **Technical Feasibility**
- **Operational Feasibility**
- **Economic Feasibility**
- **Social Feasibility**
- **Management Feasibility**
- **Legal Feasibility**
- **Time Feasibility**
- **Technical Feasibility:**

This is concerned with specifying equipment and software that will successfully satisfy the user requirement; the technical needs of the system may vary considerably, but might include: The facility to produce outputs in a given time:

- Response time under certain conditions.
- Ability to process a certain volume of transaction at a Particular speed.
- Facility to communicate data to distant location.

In examine technical feasibility; configuration of the system is given more importance than the actual makes of hardware. The configuration should give the complete picture about the system's requirement: How many workstations are required, how these units are interconnected so that they could operate and

communicate smoothly. What speeds of input and output should be achieved at particular quality of printing? This can be used as a basis for the tender document

Operational Feasibility: -

It is mainly related to human organization and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organizational structures are distributed?
- What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

Generally, project will not be rejected simply because of operational infallibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

Economical Feasibility: -

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More frequently known as cost / benefit analysis; the procedure is to determine the benefits and saving that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a change of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle**.**

Social Feasibility:

Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project accepted by the group directly affected by the proposed system change.

Management Feasibility: -

It is a determination of whether a proposed project will be acceptable to management. if does not accept a project or gives a negligible support to it; the analyst will tend to view the project as a non-feasible one**.**

Legal Feasibility: -

Legal feasibility is a determination of whether a proposed project infringes on known Acts Statutes, as well as any pending legislation. Although in some instances the project might appear sound, on closer investigation it may be found to infringe on several legal areas.

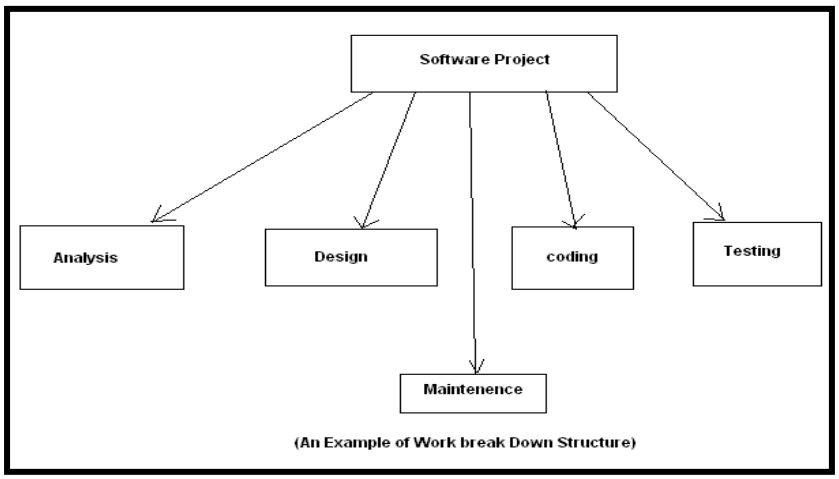
Time Feasibility:

Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame. If a project takes too much time it is likely to be rejected.

3.6 Conceptual Models

A **conceptual model** is a representation of a system, made of the composition of concepts which are used to help people know, understand, or simulate a subject the model represents. It is also a set of concepts. Some models are physical objects; for example, a project model which may be assembled, and may be made to work.

..



(An Example of Work break Down Structure)

CHAPTER-4

SYSTEM DESIGN

4.1 Basic Module

1. **Interface Module:** it contains a list of all modules. It is designed with good interface architecture so that anyone can understand it easily. User can easily select the module and perform respective task.
2. **Member Module:** User can add member details including team details. User can update the member.
3. **Team module:** it is used to add/ update details of team.
4. **Training programme module:** it is used to add programme details as team. Team is selected from team data table.
5. **Attendance Module:** It is used to add attendance status and performance rating of members.
6. **Diet plan module:** it is used to add diet plan details, update and delete unused diet plan as per requirements.

4.2 Data Design

4.2.1 Schema Design

Team

```
USE [cricket_data]
GO

/****** Object:  Table [dbo].[team]    Script Date: 23-07-2021 22:07:58 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[team](
	[team_id] [int] NOT NULL,
	[team_name] [varchar](50) NULL,
	[age_group] [varchar](50) NULL,
	[description] [varchar](200) NULL,
	[diet_plan_id] [int] NULL,
	[experience] [varchar](100) NULL,
 CONSTRAINT [PK_team] PRIMARY KEY CLUSTERED
(
	[team_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[team]  WITH CHECK ADD  CONSTRAINT
[FK_team_diet_plan] FOREIGN KEY([diet_plan_id])
REFERENCES [dbo].[diet_plan] ([diet_plan_id])
GO

ALTER TABLE [dbo].[team] CHECK CONSTRAINT [FK_team_diet_plan]
GO
```

Diet_plan

```
USE [cricket_data]
GO

/****** Object:  Table [dbo].[diet_plan]    Script Date: 23-07-2021 22:08:44 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[diet_plan](
	[diet_plan_id] [int] NOT NULL,
	[breakfast] [varchar](200) NULL,
	[Lunch] [varchar](200) NULL,
	[eve_meal] [varchar](200) NULL,
	[post_training] [varchar](200) NULL,
 CONSTRAINT [PK_training_programme] PRIMARY KEY CLUSTERED
(
	[diet_plan_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
```

GO

Attendance_performance

```
USE [cricket_data]
GO

/****** Object:  Table [dbo].[Attendance_performance]    Script Date: 23-07-2021
22:09:16 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Attendance_performance](
	[attd_id] [int] NULL,
	[prg_id] [int] NULL,
	[mem_id] [int] NULL,
	[status] [varchar](50) NULL,
	[performance] [varchar](50) NULL
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

Team Member

```
USE [cricket_data]
GO

/****** Object:  Table [dbo].[Team_member]    Script Date: 23-07-2021 22:09:26
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Team_member](
	[mem_id] [int] NOT NULL,
	[team_id] [int] NULL,
	[mname] [varchar](50) NULL,
	[email] [varchar](50) NULL,
	[address] [varchar](100) NULL,
	[contact] [varchar](10) NULL,
	[age] [int] NULL,
	[height_cm] [int] NULL,
	[weight_kg] [int] NULL,
	[role] [varchar](50) NULL,
 CONSTRAINT [PK_Team_member] PRIMARY KEY CLUSTERED
(
	[mem_id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Team_member]  WITH CHECK ADD  CONSTRAINT
[FK_Team_member_team] FOREIGN KEY([team_id])
REFERENCES [dbo].[team] ([team_id])
GO

ALTER TABLE [dbo].[Team_member] CHECK CONSTRAINT
[FK_Team_member_team]
GO
```

Training_programme

```
USE [cricket_data]
GO

/****** Object:  Table [dbo].[training_programme]    Script Date: 23-07-2021
22:09:48 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[training_programme](
	[prg_id] [int] NOT NULL,
	[team_id] [int] NULL,
	[tdate] [date] NULL,
	[time] [time](7) NULL,
	[excercise_details] [varchar](200) NULL,
 CONSTRAINT [PK_training_programme_1] PRIMARY KEY CLUSTERED
(
	[prg_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

```
SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[training_programme]  WITH CHECK ADD  CONSTRAINT
[FK_training_programme_team] FOREIGN KEY([team_id])
REFERENCES [dbo].[team] ([team_id])
GO

ALTER TABLE [dbo].[training_programme] CHECK CONSTRAINT
[FK_training_programme_team]
GO
```

4.2.2 Data Integrity and Constraints

Integrity Constraints:

Data integrity constraints provide a means of ensuring that change made to the database by authorized users don't result in loss of data consistency. Thus, integrity constraints guards against accidental damage to the database.

In relational database the integrity constraints ate of follows

Key declaration –

The field for which we have mo two rows with same value. It lets the database with legal insertion and updates.

Form of Relationship – The relation ship between two entities in database. These relationships are of the form.

- **One-to-One**
- **One-to-Many**
- **Many-to-One**
- **Many-to-Many**

Relational Integrity

It is often seen that a value which appears in one relation for a given set of attributes also appears for certain set of attributes in another relation. This is called referential integrity.

Referential Integrity in SQL

Primary Key -

The primary key clause of the create tuple student is the attribute which contains its uniqueness. We can access any other field using primary key.

Foreign key –

Foreign key is the concept to relate two tuples in such a way that their integrity and constraints could not be lost.

Some item we see that a primary key of one tuple is candidate key of another tuple.

Unique Key –

It same as primary key except that unique key can contain a null value which is not possible in the case of primary key.

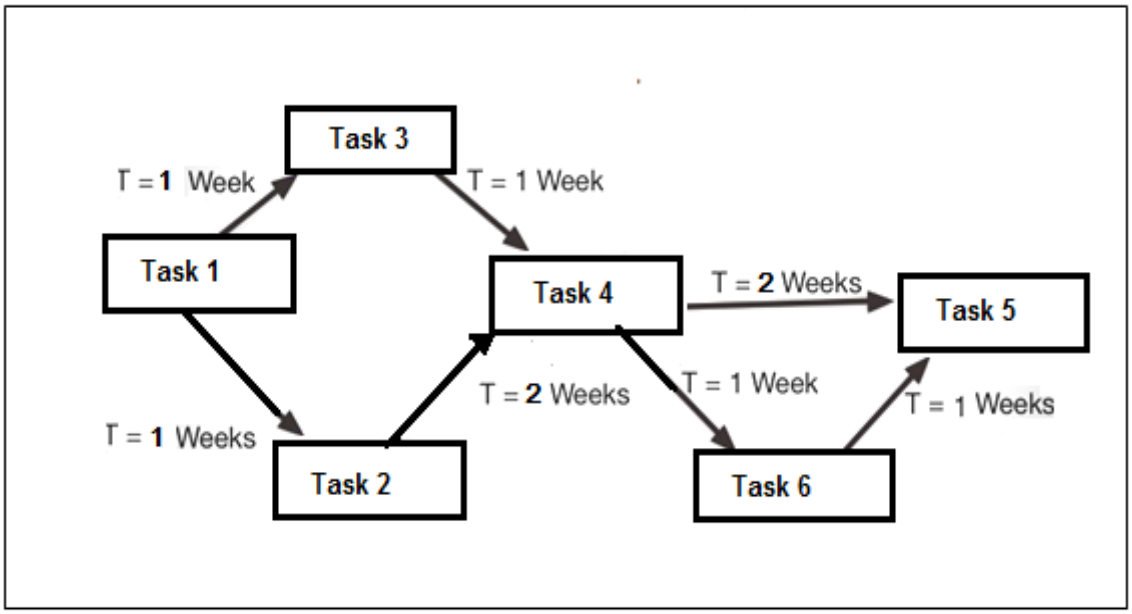
4.3 Procedural Design

4.3.1 Logic Diagrams: -

PERT CHART

A PERT **(Program Evaluation and Review Technique)** chart is a network of boxes (or circles) and arrows. There are different variations of PERT charts. The boxes in PERT Chart can be decorated with starting and ending dates for activities; the arrows help in computing the earliest possible starting dates for the boxes at their heads. Some boxes can be designated as milestone.

The Chart below shows that the path through the project that consists of the "design", "build code generator", and "integration and testing" activities is the critical path for the project. Any delay in any activity in this path will cause a delay in the entire project.

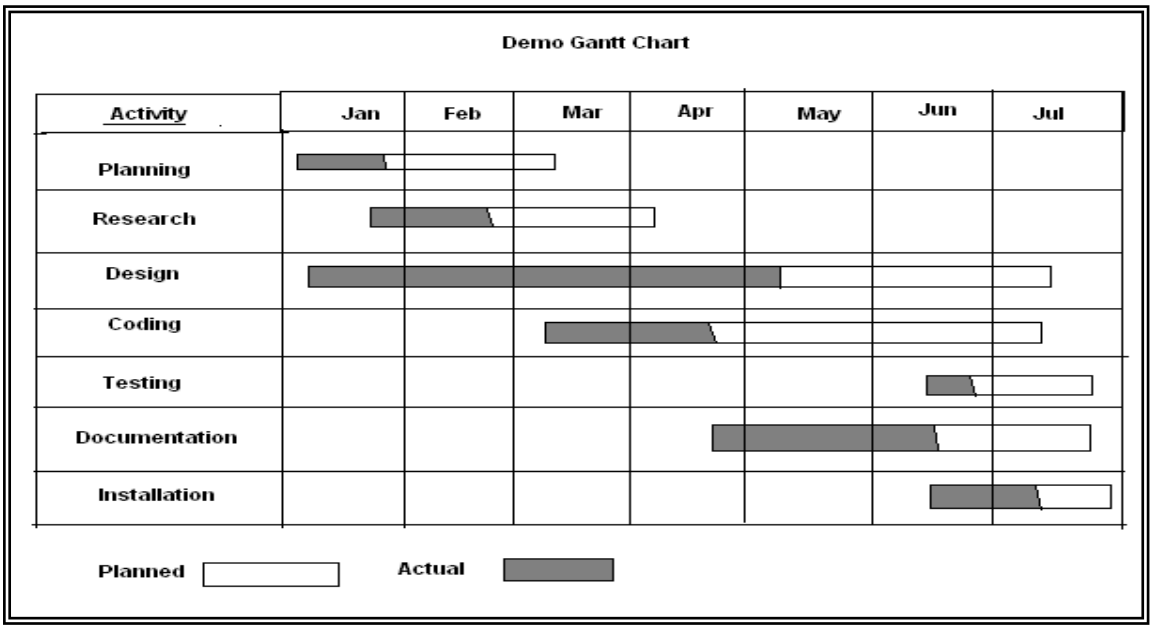


GANTT CHART

Gantt charts are a project control technique that can be used for several purposes, including scheduling, budgeting, and resource planning. A Gantt chart is a bar chart, with each bar representing an activity. The bars are drawn against a time Line. The length of each bar is proportional to the length of time planned for the activity.

A Gantt chart helps in scheduling the activities of a project, but it does not help in identifying them. Gantt charts take different forms depending on their intended use. They are best for resource scheduling.

The Diagram of Gantt Chart for this project is given on the following page.



4.3.2 Data Structures

Team

Attributes	Datatype	Size	Constraints	description
Team_id	Int		Primary Key	This is team id number
Team_name	Varchar	50	Not Null	Team name
Age_group	Varchar	50	Not Null	Team for which Age group
Descr	Varchar	200		Description
Diet_plan_id	Int		Foreign key(diet_plan)	Diet_plan_id selected for team.
Experience	Varchar	100	Not Null	Experience details

Team_Member

Attributes	Datatype	Size	Constraints	Description
Mem_id	Int		Primary key	Member id
Team_id	Int		Foreign key (Team)	Team id number
mname	Varchar	50	Not Null	Member name
Email	Varchar	50	Not Null	Email ID
Address	Varchar	100	Not Null	Address
Contact	Varchar	10	Not Null	Contact number
Age	Int		Not Null	Age of member
Height	Int		Not Null	Height of member
Weight	Int		Not Null	Weight of member
Role	Varchar	50		Roles in team

Diet_plan

Attributes	Datatype	Size	Constraints	Description
Diet_plan__id	Int		Primary key	Diet plan ID
Breakfast	Varchar	200	Not Null	Breakfast diet
Lunch	Varchar	200	Not Null	Lunch diet details
Eve_meal	Varchar	200	Not Null	Evening meal
Post_training	Varchar	200	Not Null	Post training meal

Training_programme

Attributes	Datatype	Size	Constraints	Description
Prg_id	Int		Primary key	Programme ID
Team_id	Int		Foreign key (team)	Team id
tdate	Date		Not Null	Training date
Time	Date/Time		Not Null	Timing
Exercise_details	Varchar	200	Not Null	Exercise details
			Not Null	

Attendance_performance

Attributes	Datatype	Size	Constraints	Description
Attd_id	Int		Primary key	Attendance id
Prg_id	Int		Foreign key (training programme)	Programme id
Mem_id	Int		Foreign key	Member ID
status	Varchar		Not Null	Attend status(present/Absent)
Performance	Varchar		Not Null	Performance details (Excellent/ Very good/good/poor)

4.3.3 Algorithms Design

Data Flow Digram

An arrow represents data flow; it represents the path over which data travels in the system. A data flow can move between processes, flow into or out of data stores, to and from external entities.

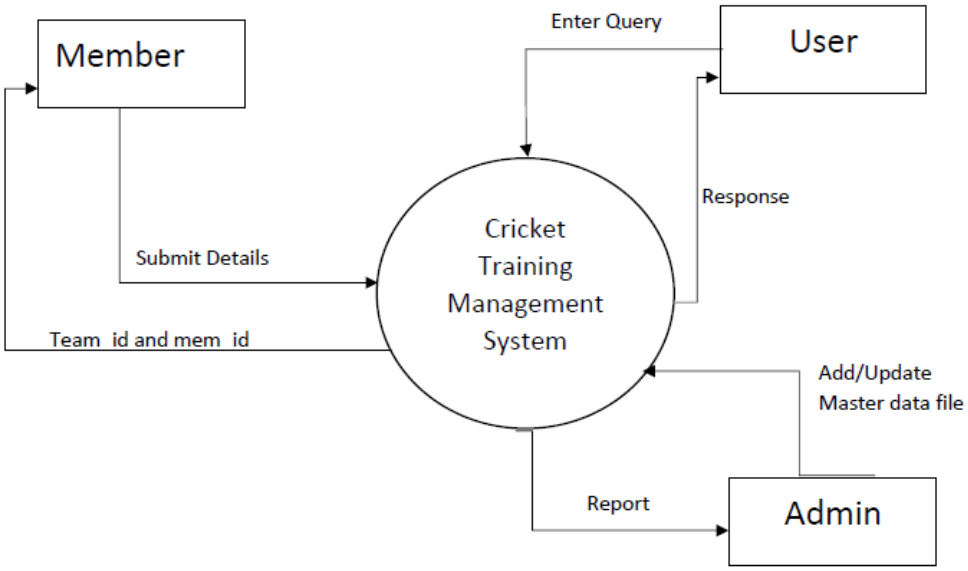
0 LEVEL DFD: -

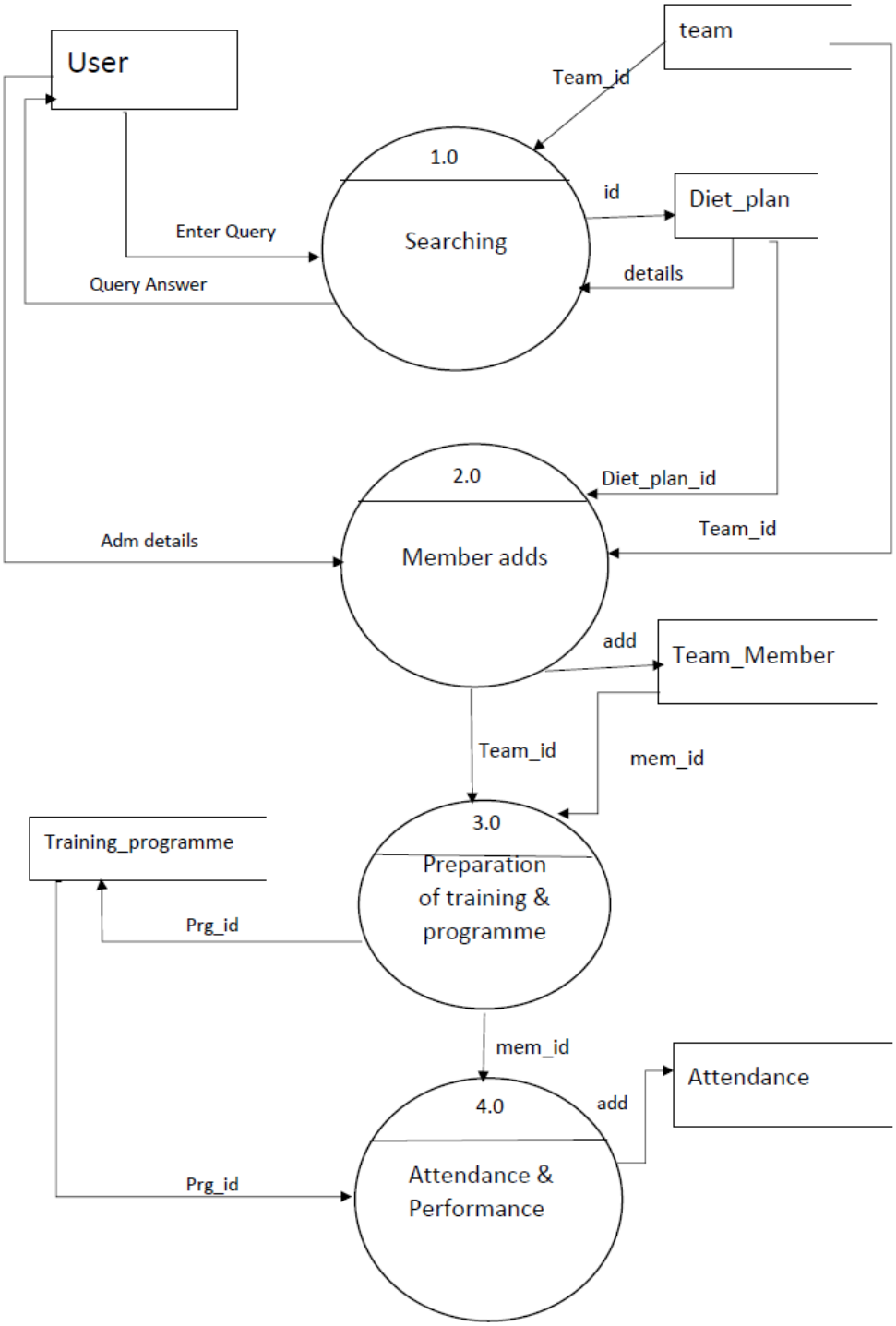
This is the context level DFD of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The purposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system

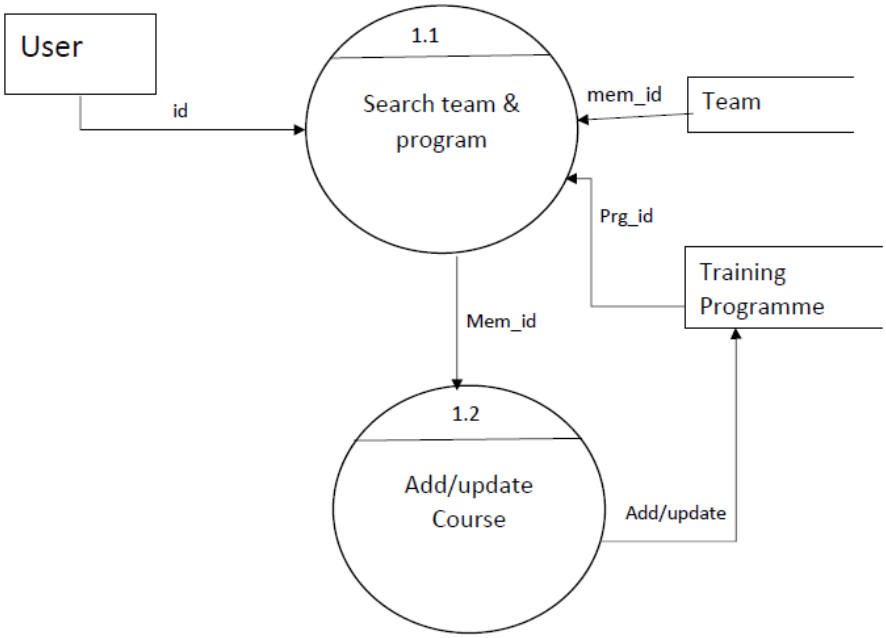
DFD level 0:

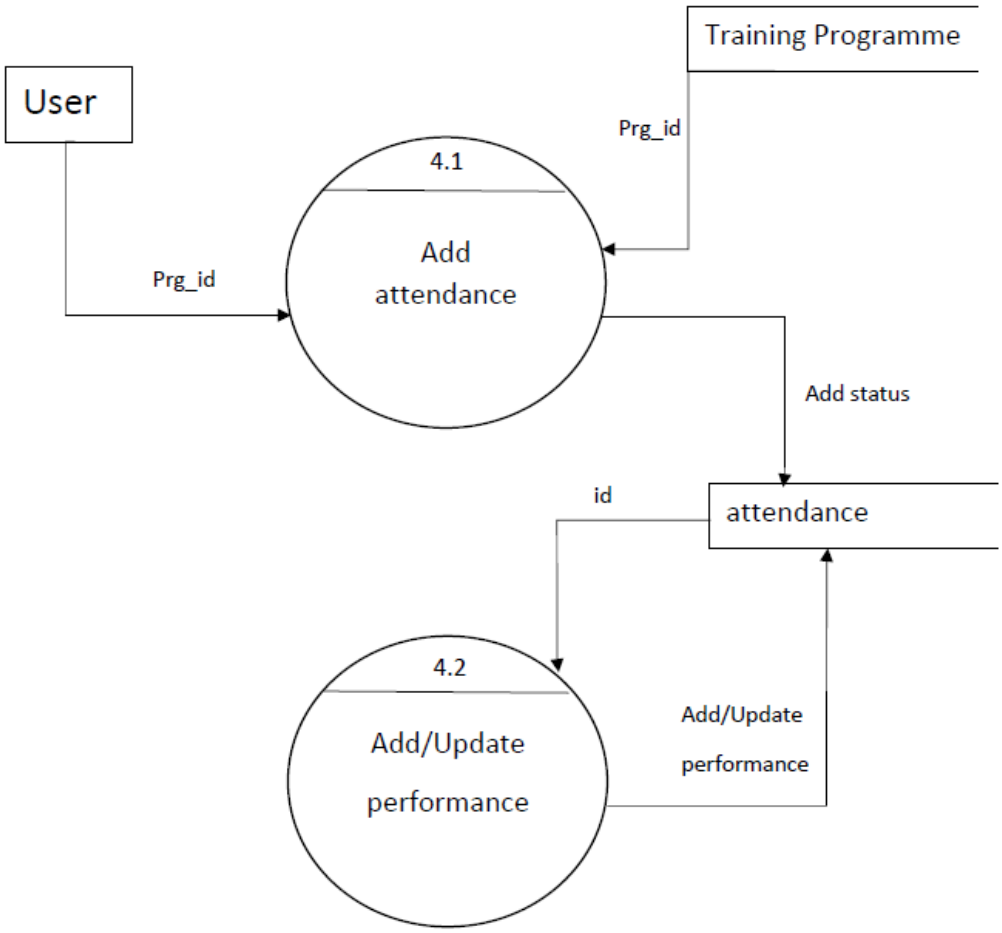
This is the context level DFD of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The purposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system

DFD level 0:

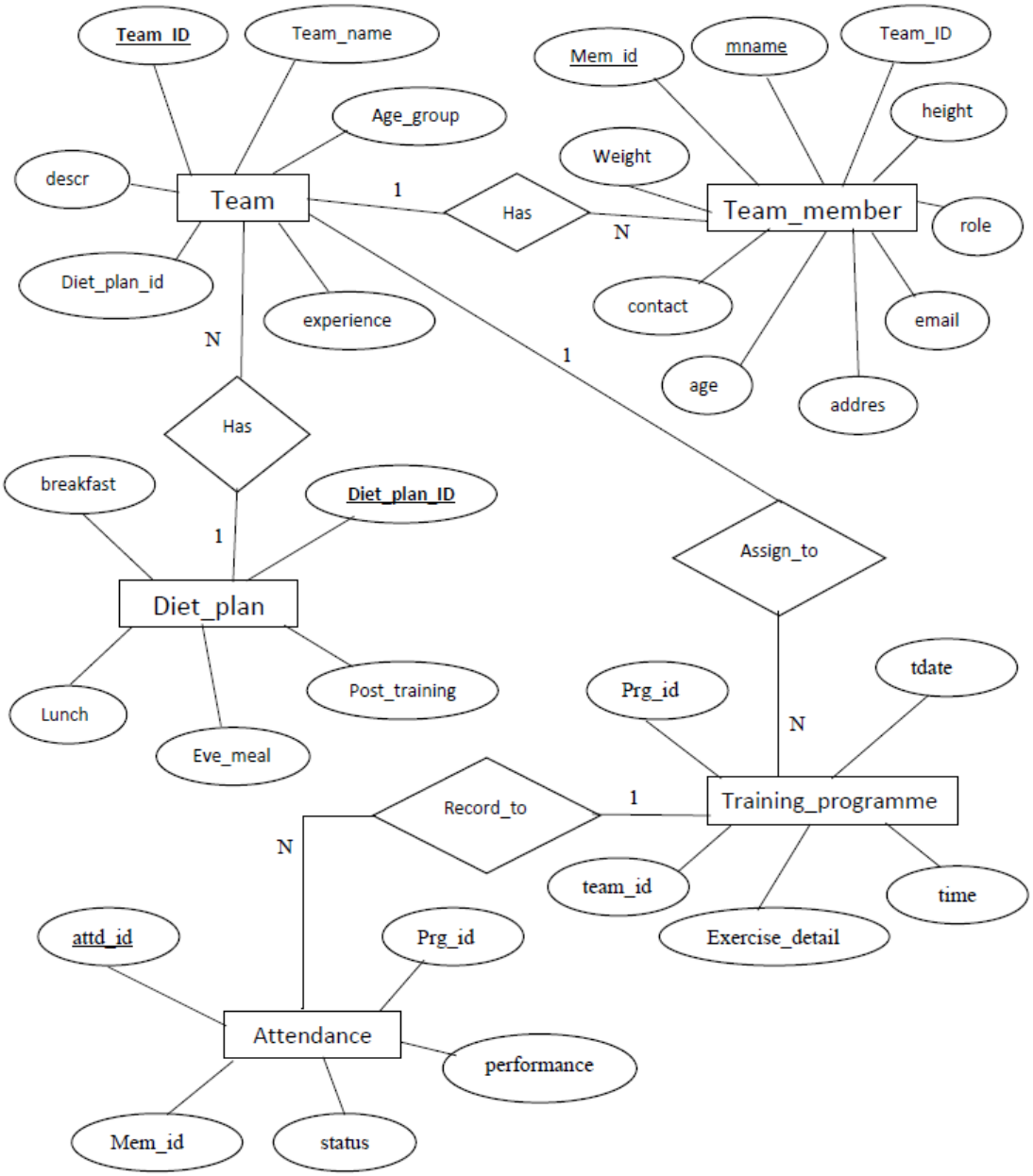

Member
Enter Query
User
Response
Submit Details
Cricket
Training
Management
System
Team id and mem id
Add/Update
Master data file
Report
Admin


User
team
Team_id
1.0
Searching
id
Diet_plan
Enter Query
Query Answer
details
2.0
Member adds
Diet_plan_id
Team_id
Adm details
add
Team_Member
Team_id
mem_id
3.0
Preparation of training & programme
Training_programme
Prg_id
mem_id
4.0
Attendance & Performance
add
Attendance
Prg_id

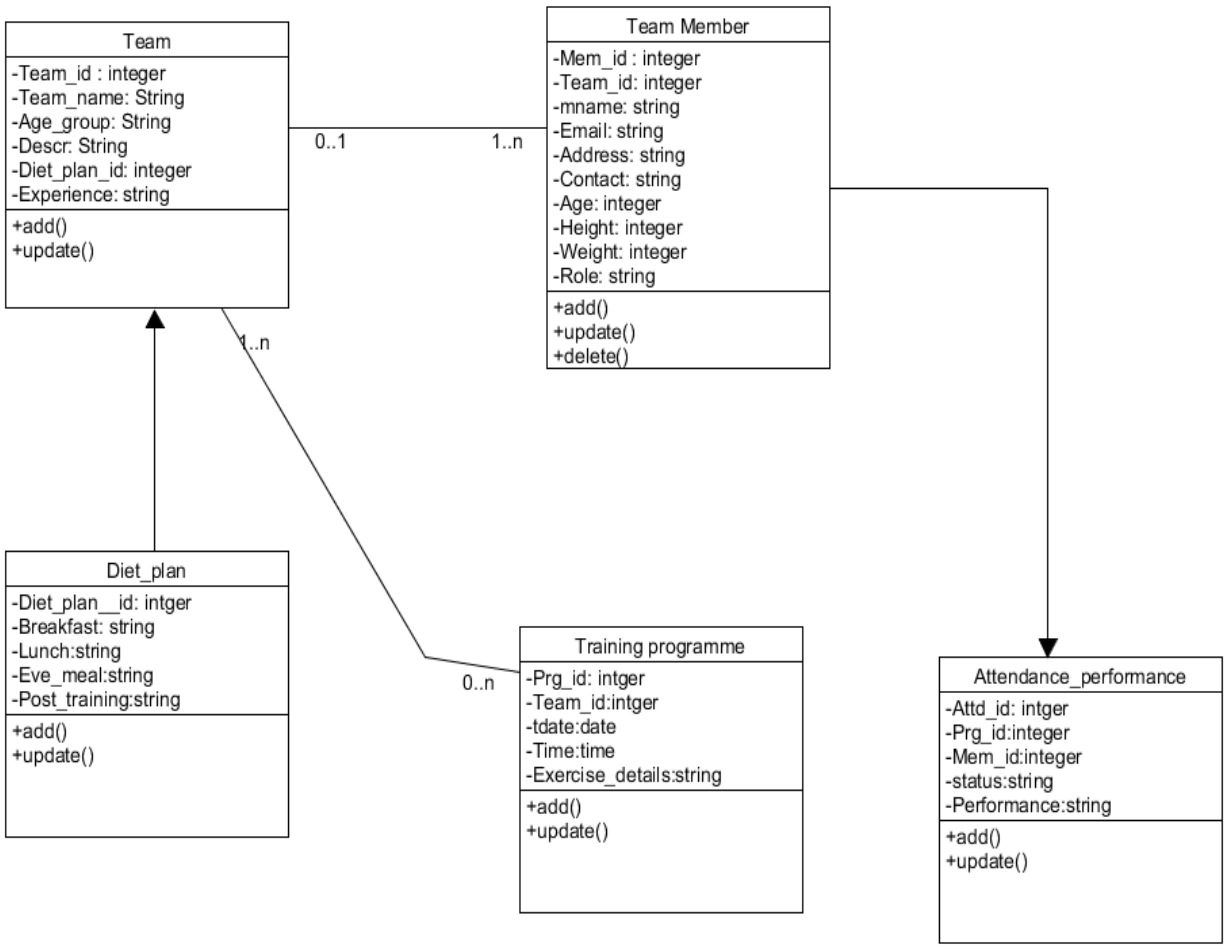

User
id
1.1
Search team &
program
mem_id
Team
Prg_id
Training
Programme
Mem_id
1.2
Add/update
Course
Add/update


Training Programme
User
Prg_id
4.1
Add
attendance
Prg_id
Add status
id
attendance
4.2
Add/Update
performance
Add/Update
performance

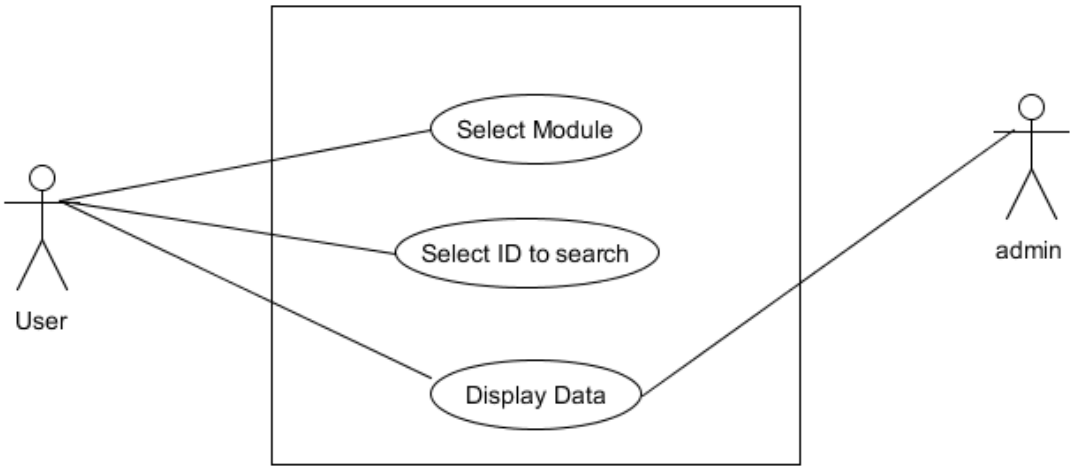
3.10 ER Diagram

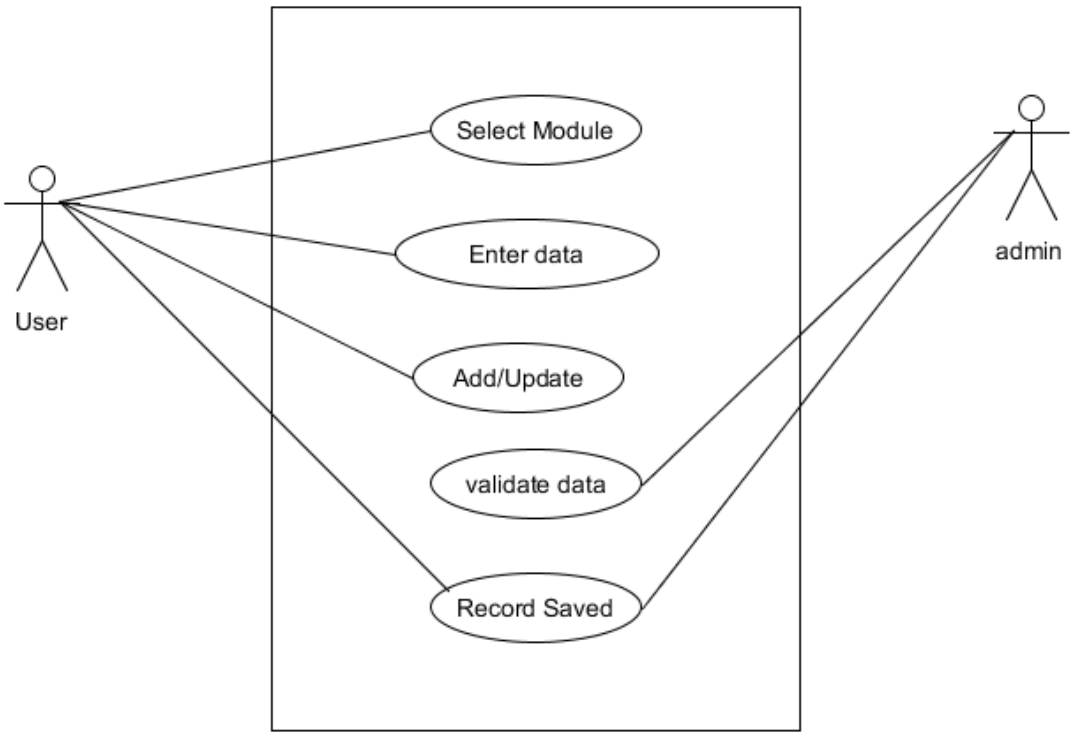


Class Diagram

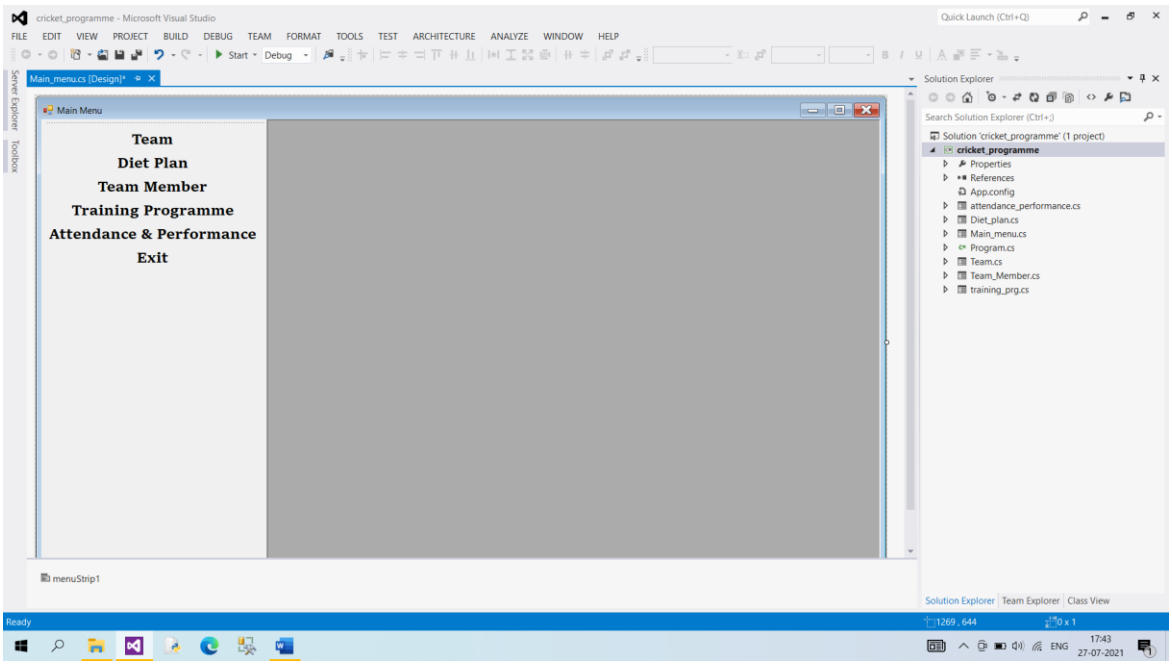


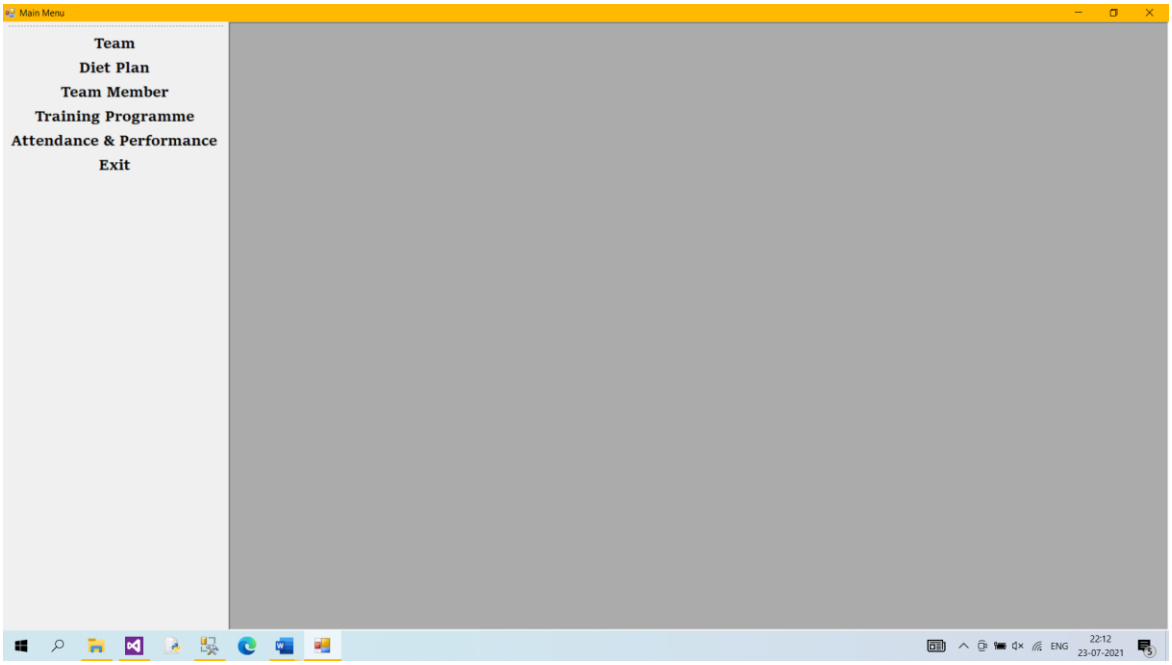
Use Case

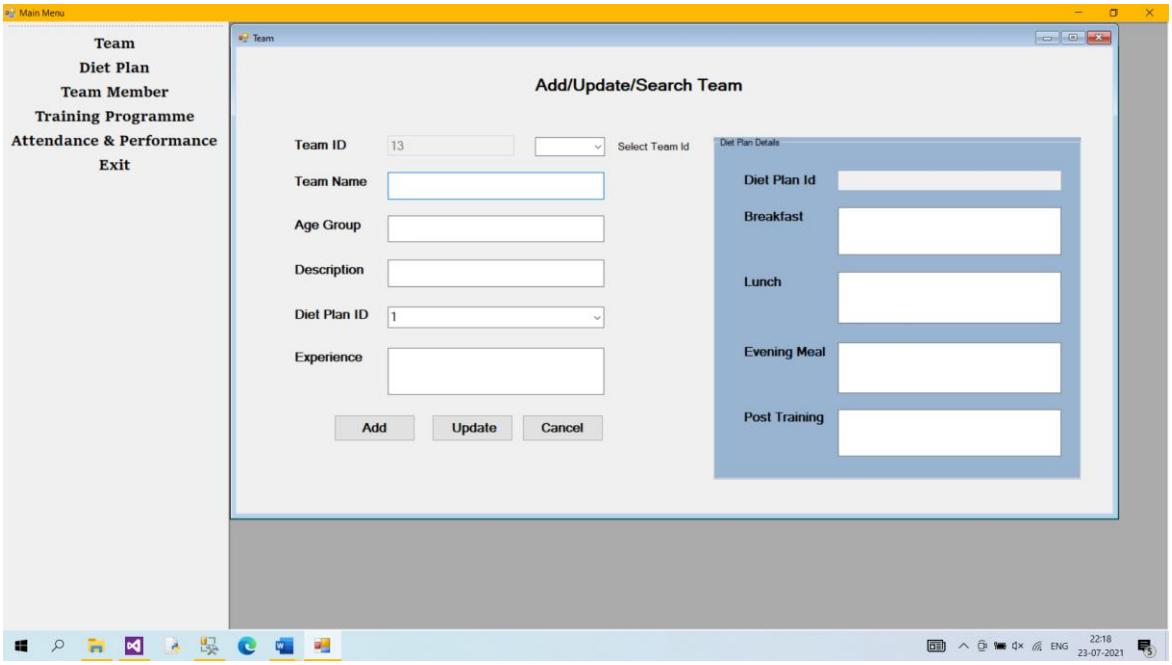

Select Module
Select ID to search
Display Data
User
admin

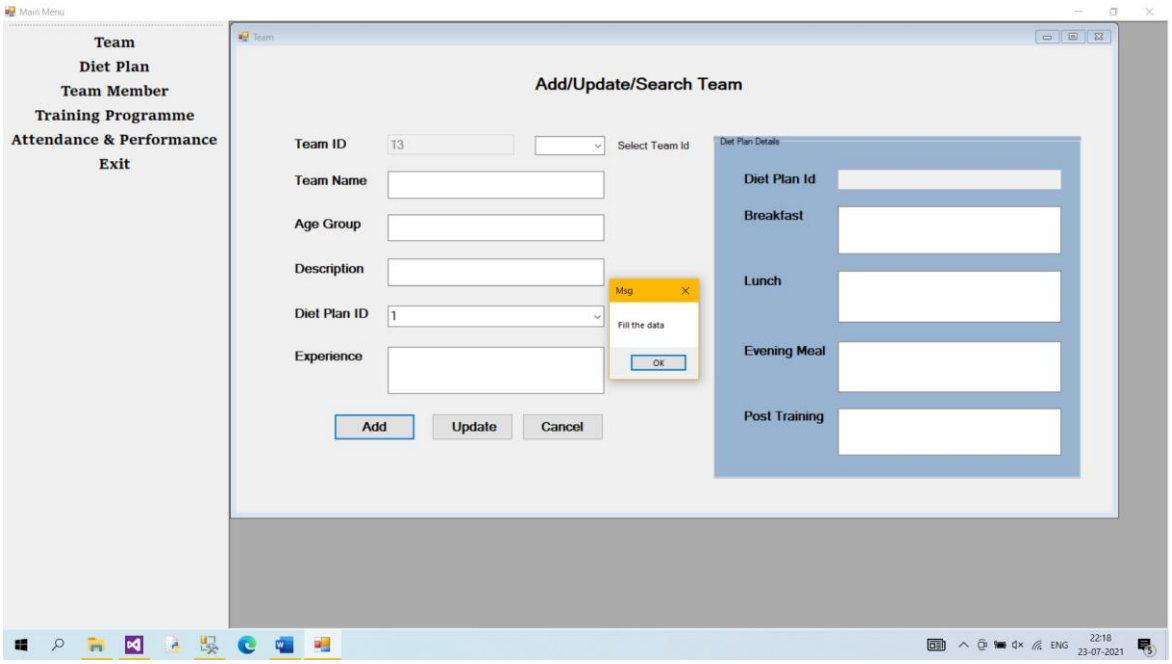

Select Module
Enter data
Add/Update
validate data
Record Saved
User
admin

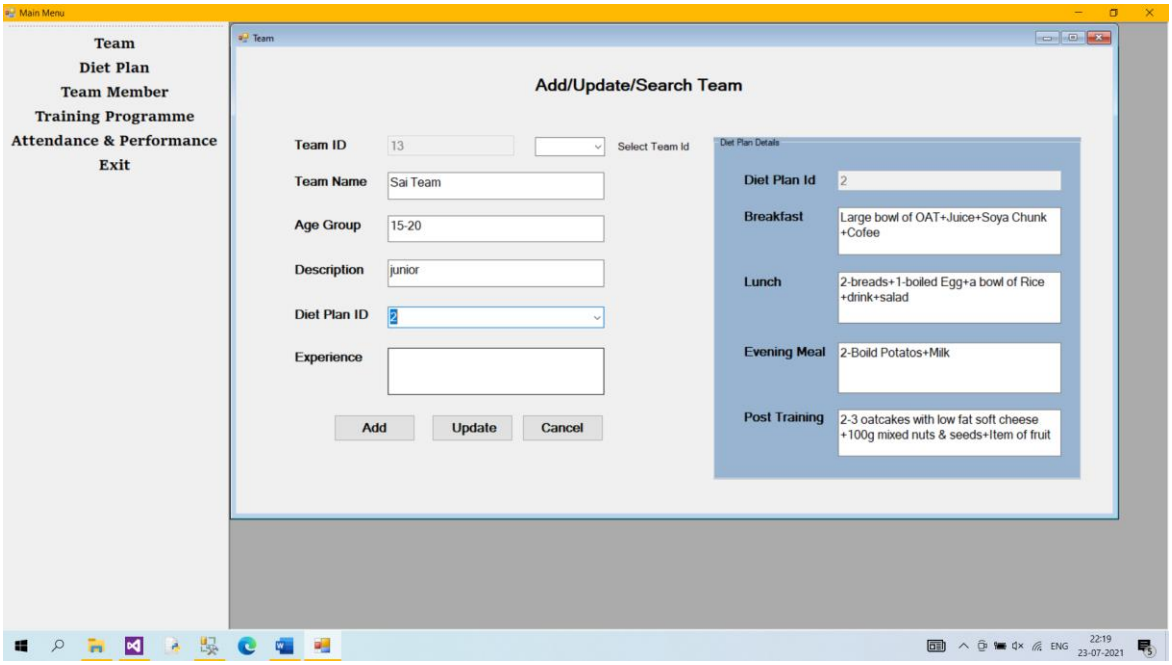
4.4 User Interface Design

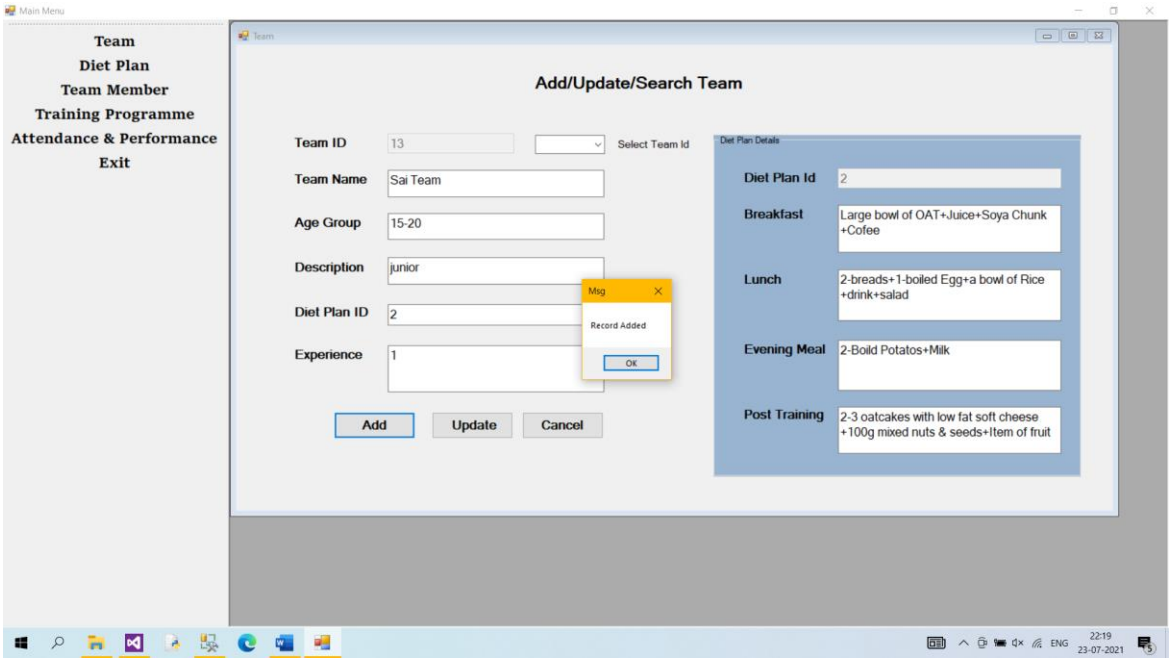


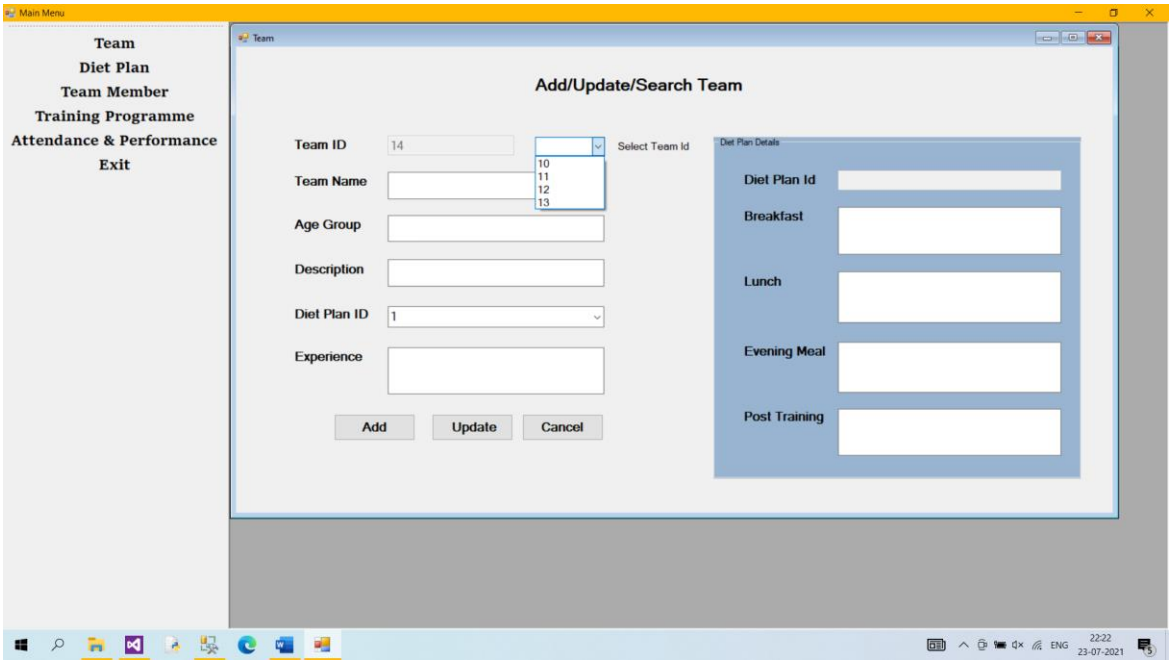

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
ENG
22:12
23-07-2021

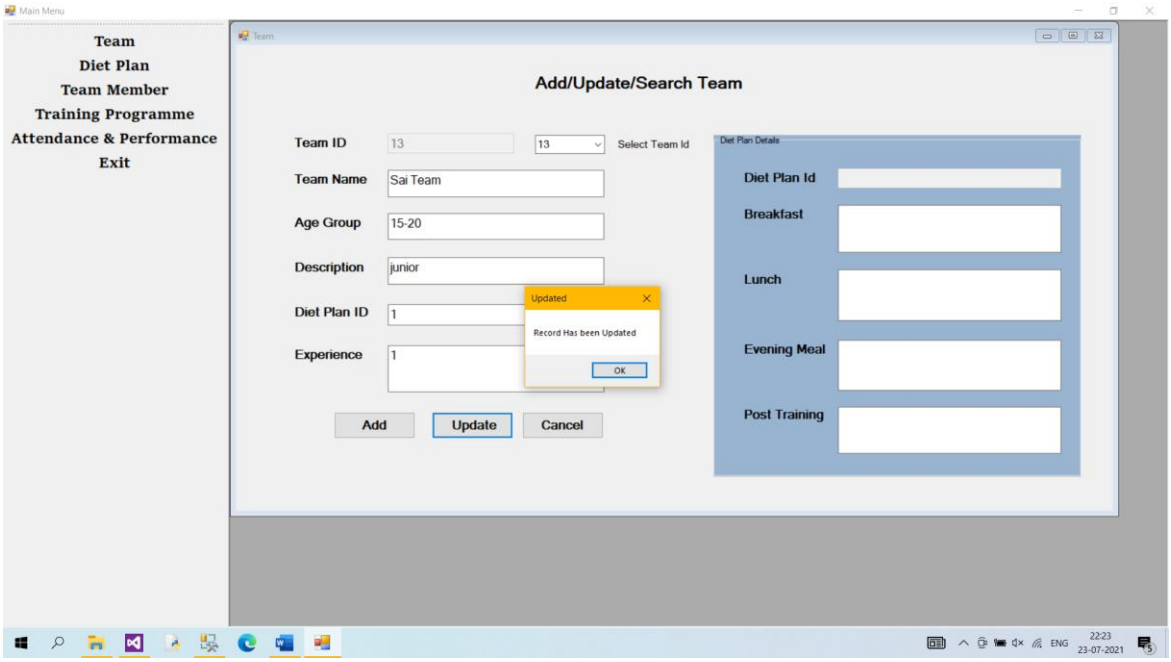

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team
Add/Update/Search Team
Team ID
13
Select Team Id
Team Name
Age Group
Description
Diet Plan ID
1
Experience
Add
Update
Cancel
Diet Plan Details
Diet Plan Id
Breakfast
Lunch
Evening Meal
Post Training
ENG
22:18
23-07-2021

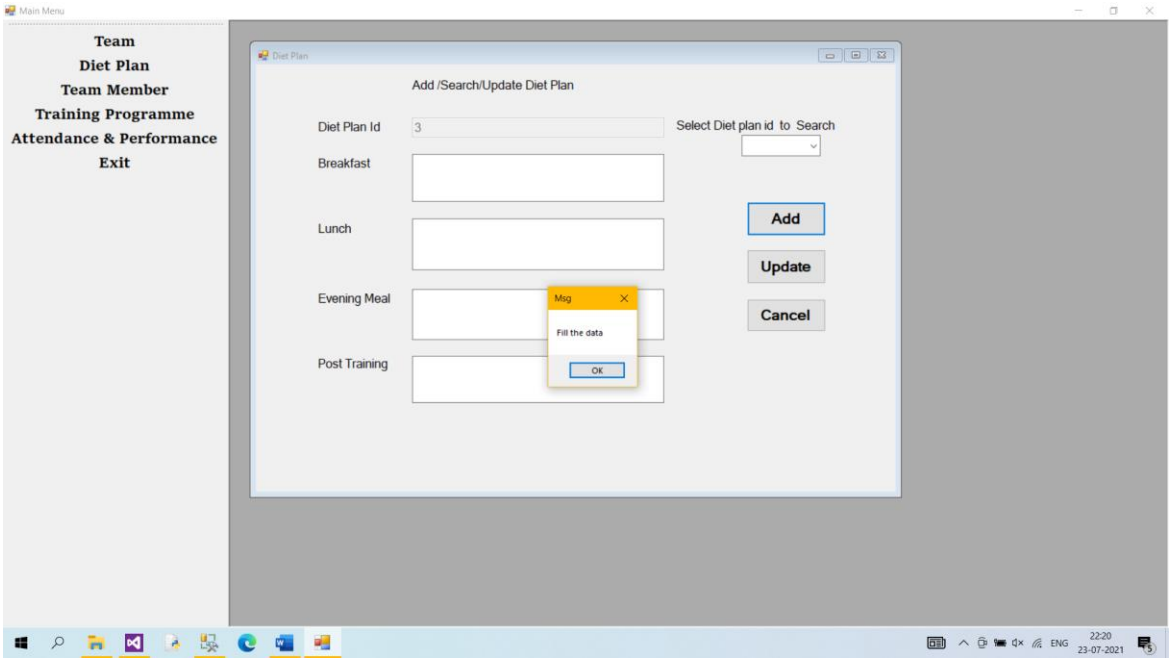

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team
Add/Update/Search Team
Team ID
13
Select Team Id
Team Name
Age Group
Description
Diet Plan ID
1
Experience
Add
Update
Cancel
Msg
Fill the data
OK
Diet Plan Details
Diet Plan Id
Breakfast
Lunch
Evening Meal
Post Training
ENG
22:18
23-07-2021

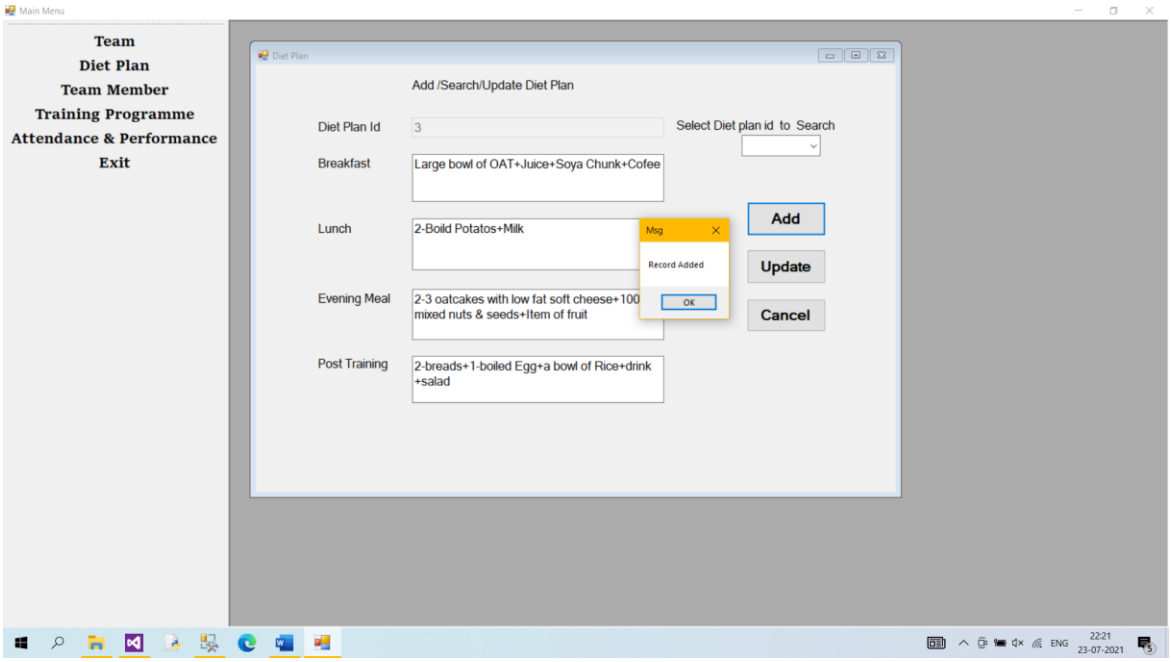

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team
Add/Update/Search Team
Team ID
13
Select Team Id
Team Name
Sai Team
Age Group
15-20
Description
junior
Diet Plan ID
2
Experience
Add
Update
Cancel
Diet Plan Details
Diet Plan Id
2
Breakfast
Large bowl of OAT+Juice+Soya Chunk +Cofee
Lunch
2-breads+1-boiled Egg+a bowl of Rice +drink+salad
Evening Meal
2-Boild Potatos+Milk
Post Training
2-3 oatcakes with low fat soft cheese +100g mixed nuts & seeds+Item of fruit
ENG
22:19
23-07-2021

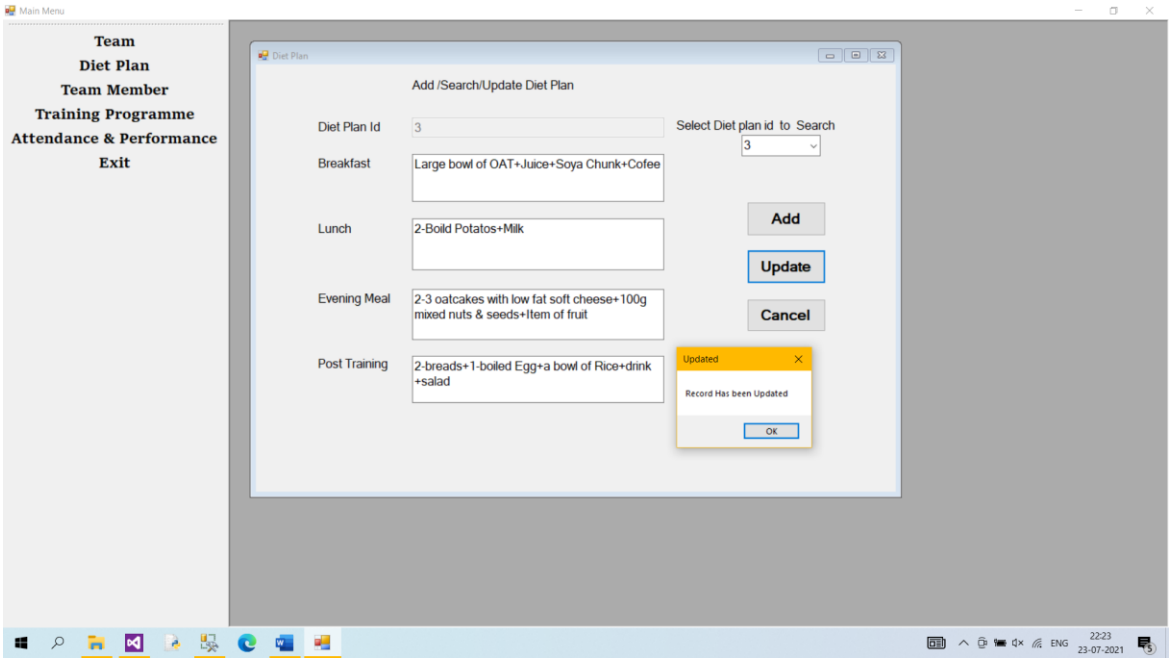

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team
Add/Update/Search Team
Team ID
13
Select Team Id
Team Name
Sai Team
Age Group
15-20
Description
junior
Diet Plan ID
2
Experience
1
Add
Update
Cancel
Msg
Record Added
OK
Diet Plan Details
Diet Plan Id
2
Breakfast
Large bowl of OAT+Juice+Soya Chunk +Cofee
Lunch
2-breads+1-boiled Egg+a bowl of Rice +drink+salad
Evening Meal
2-Boild Potatos+Milk
Post Training
2-3 oatcakes with low fat soft cheese +100g mixed nuts & seeds+Item of fruit
ENG
22:19
23-07-2021

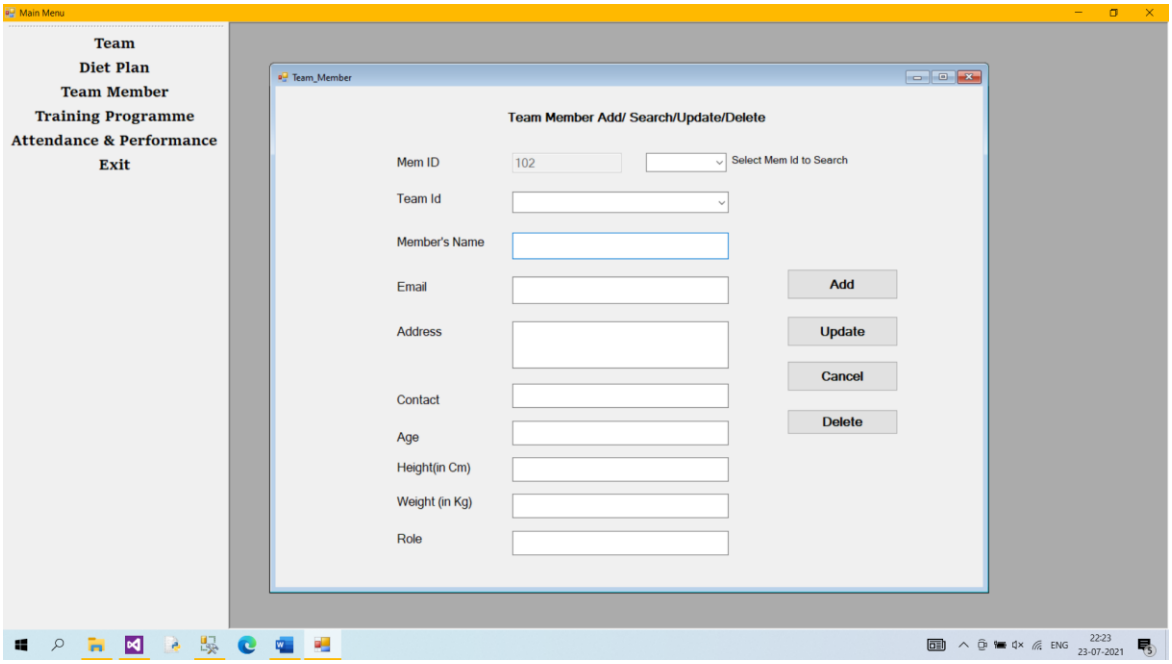

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team
Add/Update/Search Team
Team ID
14
Select Team Id
10
11
12
13
Team Name
Age Group
Description
Diet Plan ID
1
Experience
Add
Update
Cancel
Diet Plan Details
Diet Plan Id
Breakfast
Lunch
Evening Meal
Post Training
ENG
22:22
23-07-2021

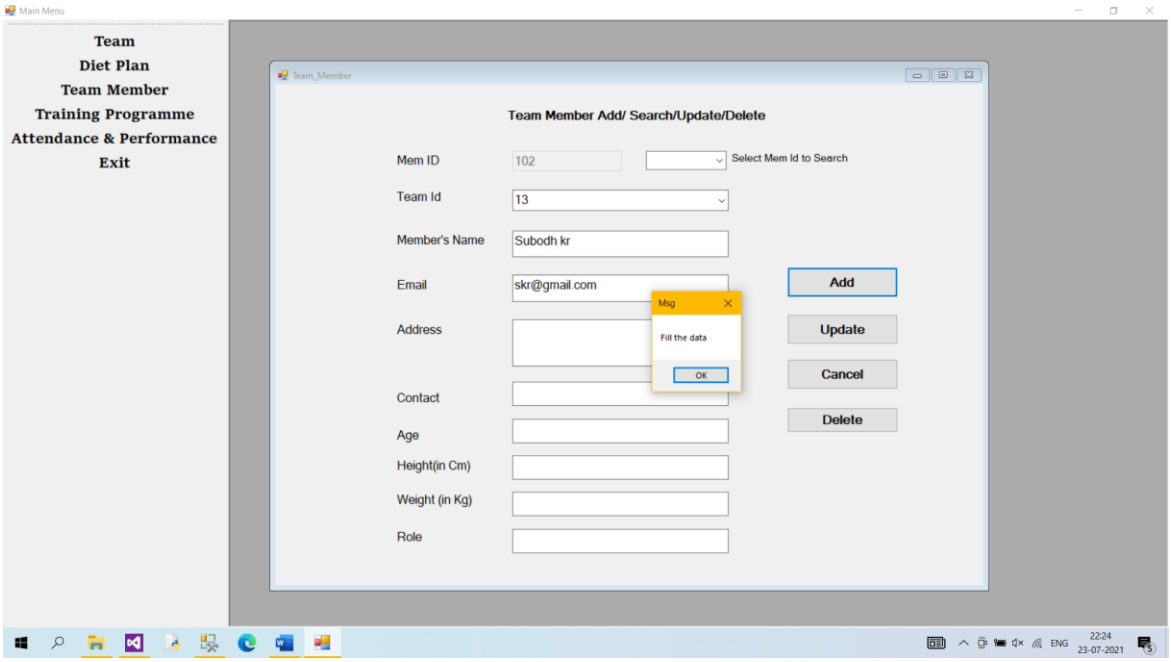

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team
Add/Update/Search Team
Team ID
13
13
Select Team Id
Team Name
Sai Team
Age Group
15-20
Description
junior
Diet Plan ID
1
Experience
1
Add
Update
Cancel
Updated
Record Has been Updated
OK
Diet Plan Details
Diet Plan Id
Breakfast
Lunch
Evening Meal
Post Training
ENG
22:23
23-07-2021

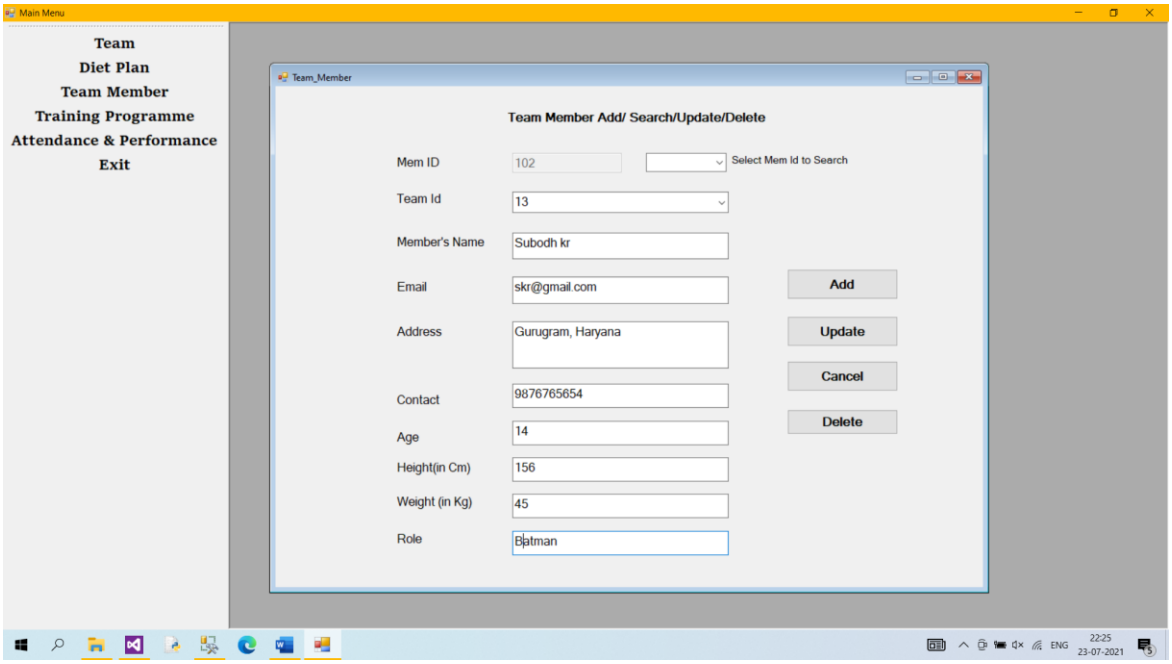

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Diet Plan
Add /Search/Update Diet Plan
Diet Plan Id
3
Select Diet plan id to Search
Breakfast
Lunch
Evening Meal
Post Training
Add
Update
Cancel
Msg
Fill the data
OK
ENG
22:20
23-07-2021

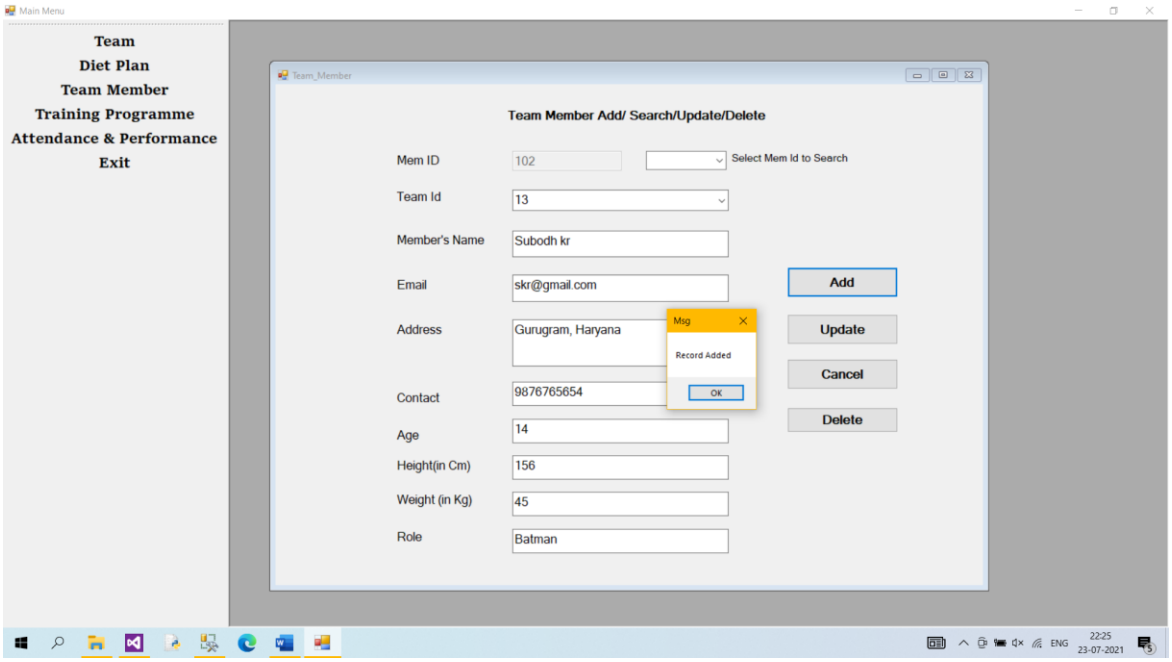

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Diet Plan
Add /Search/Update Diet Plan
Diet Plan Id
3
Select Diet plan id to Search
Breakfast
Large bowl of OAT+Juice+Soya Chunk+Cofee
Lunch
2-Boild Potatos+Milk
Evening Meal
2-3 oatcakes with low fat soft cheese+100 mixed nuts & seeds+Item of fruit
Post Training
2-breads+1-boiled Egg+a bowl of Rice+drink +salad
Add
Update
Cancel
Msg
Record Added
OK
ENG
22:21
23-07-2021

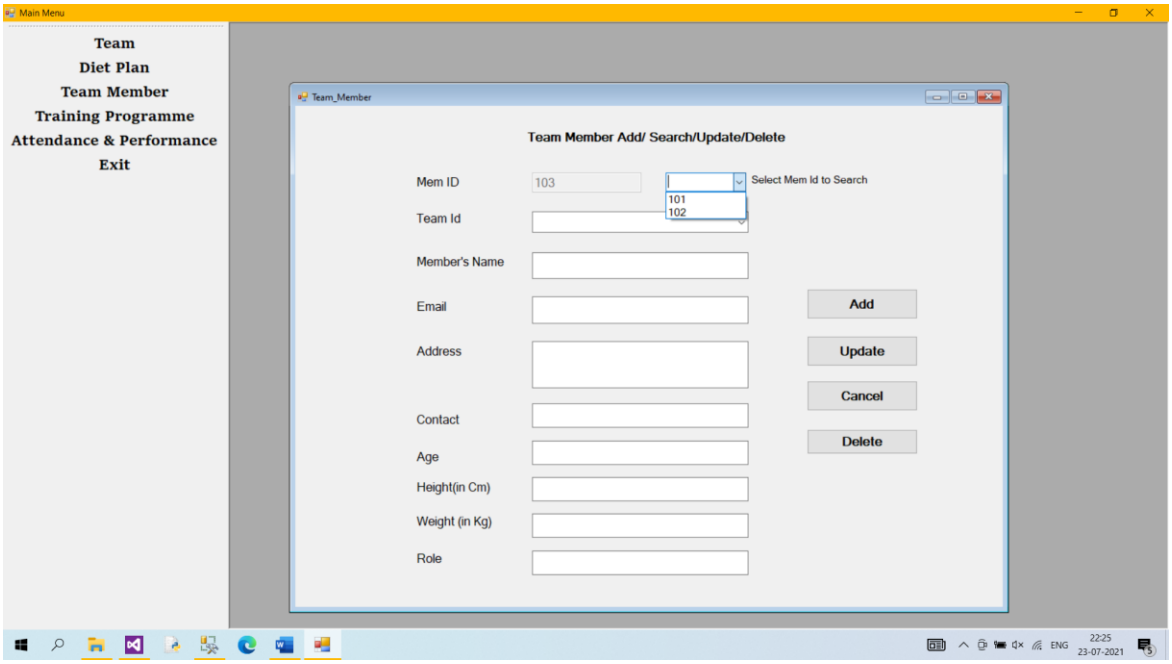

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Diet Plan
Add /Search/Update Diet Plan
Diet Plan Id
3
Select Diet plan id to Search
3
Breakfast
Large bowl of OAT+Juice+Soya Chunk+Cofee
Lunch
2-Boild Potatos+Milk
Add
Update
Cancel
Evening Meal
2-3 oatcakes with low fat soft cheese+100g mixed nuts & seeds+Item of fruit
Post Training
2-breads+1-boiled Egg+a bowl of Rice+drink +salad
Updated
Record Has been Updated
OK
ENG
22:23
23-07-2021

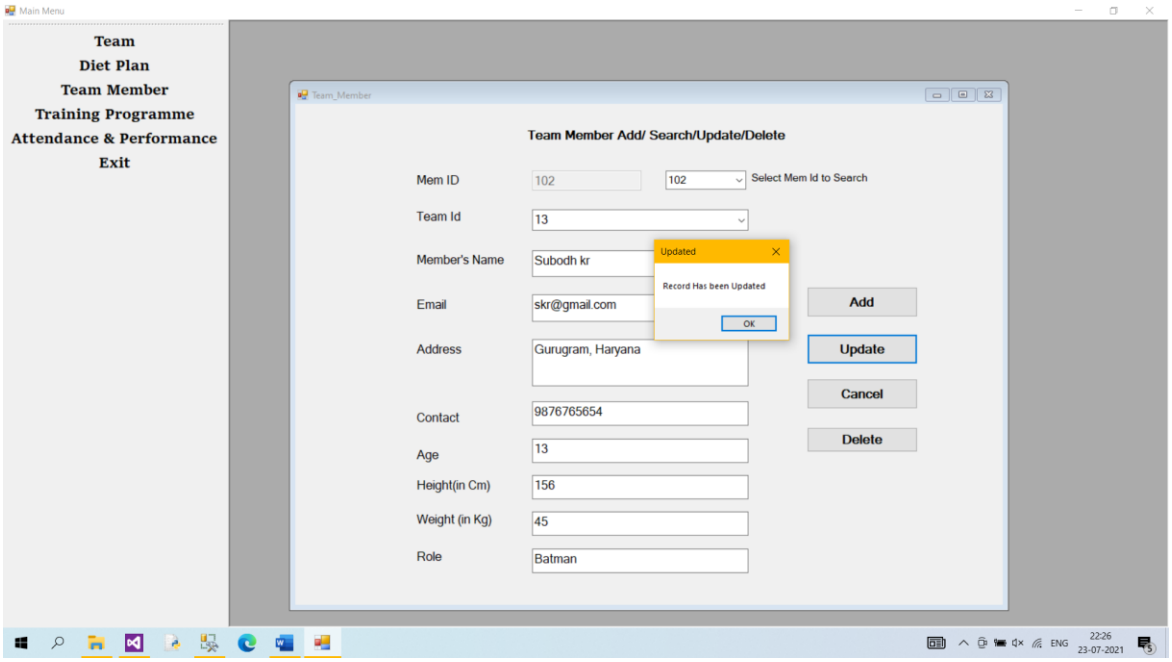

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team_Member
Team Member Add/ Search/Update/Delete
Mem ID
102
Select Mem Id to Search
Team Id
Member's Name
Email
Address
Contact
Age
Height(in Cm)
Weight (in Kg)
Role
Add
Update
Cancel
Delete
ENG
22:23
23-07-2021

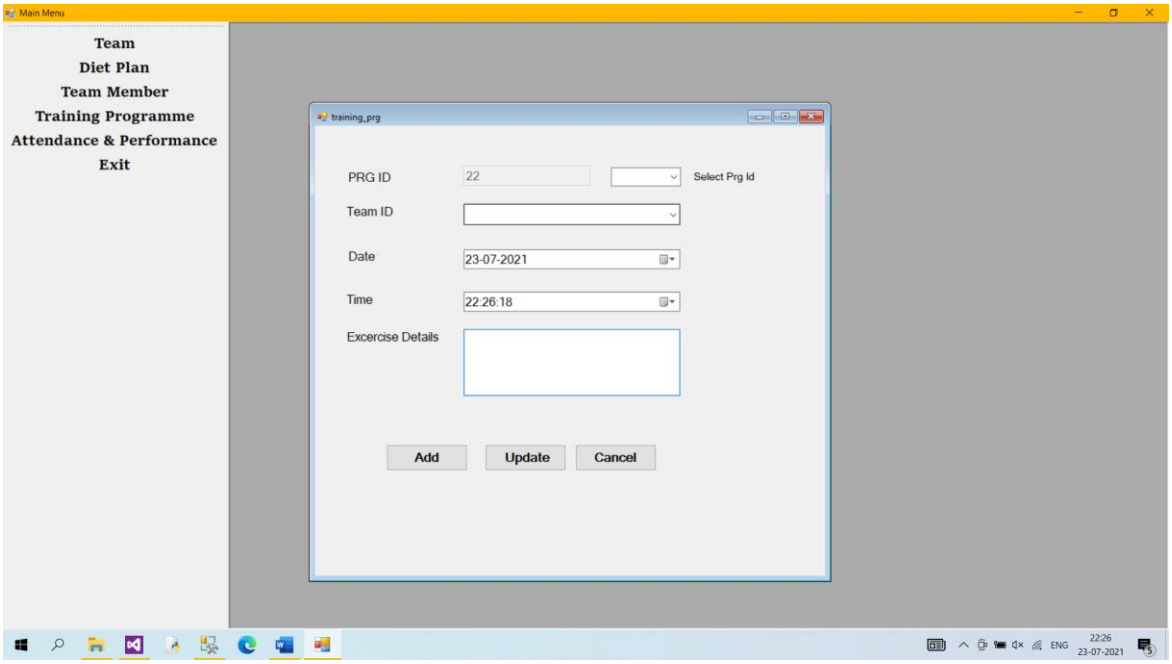

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team_Member
Team Member Add/ Search/Update/Delete
Mem ID
102
Select Mem Id to Search
Team Id
13
Member's Name
Subodh kr
Email
skr@gmail.com
Address
Contact
Age
Height(in Cm)
Weight (in Kg)
Role
Add
Update
Cancel
Delete
Msg
Fill the data
OK
ENG
22:24
23-07-2021

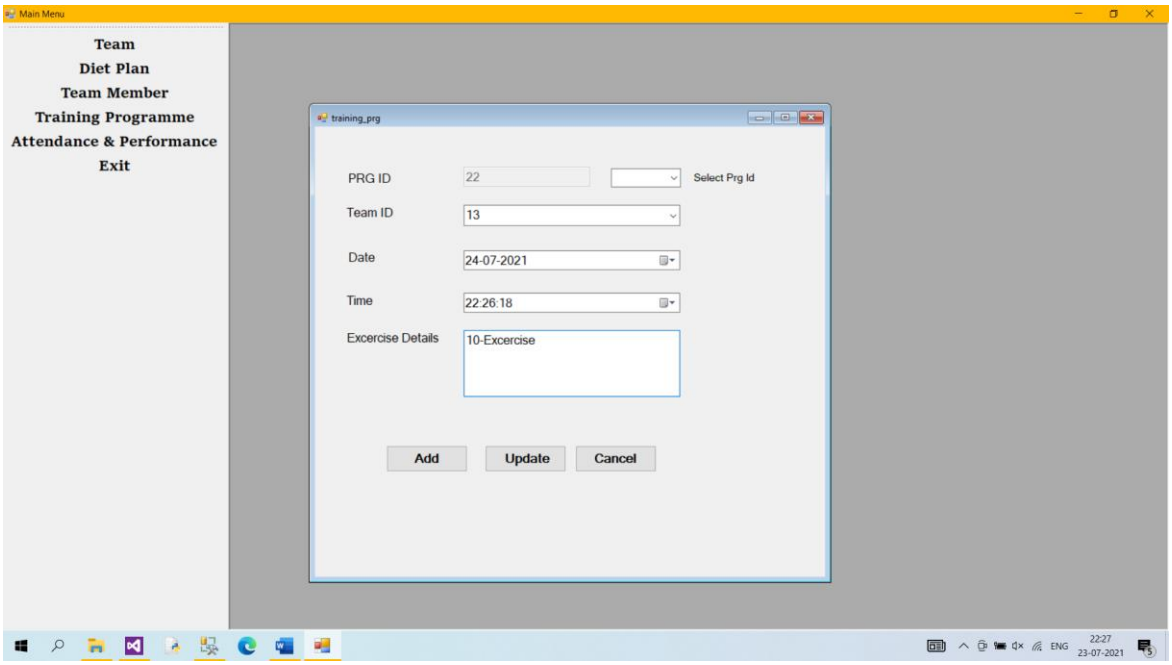

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team_Member
Team Member Add/ Search/Update/Delete
Mem ID
102
Select Mem Id to Search
Team Id
13
Member's Name
Subodh kr
Email
skr@gmail.com
Address
Gurugram, Haryana
Contact
9876765654
Age
14
Height(in Cm)
156
Weight (in Kg)
45
Role
Batman
Add
Update
Cancel
Delete
ENG
22:25
23-07-2021

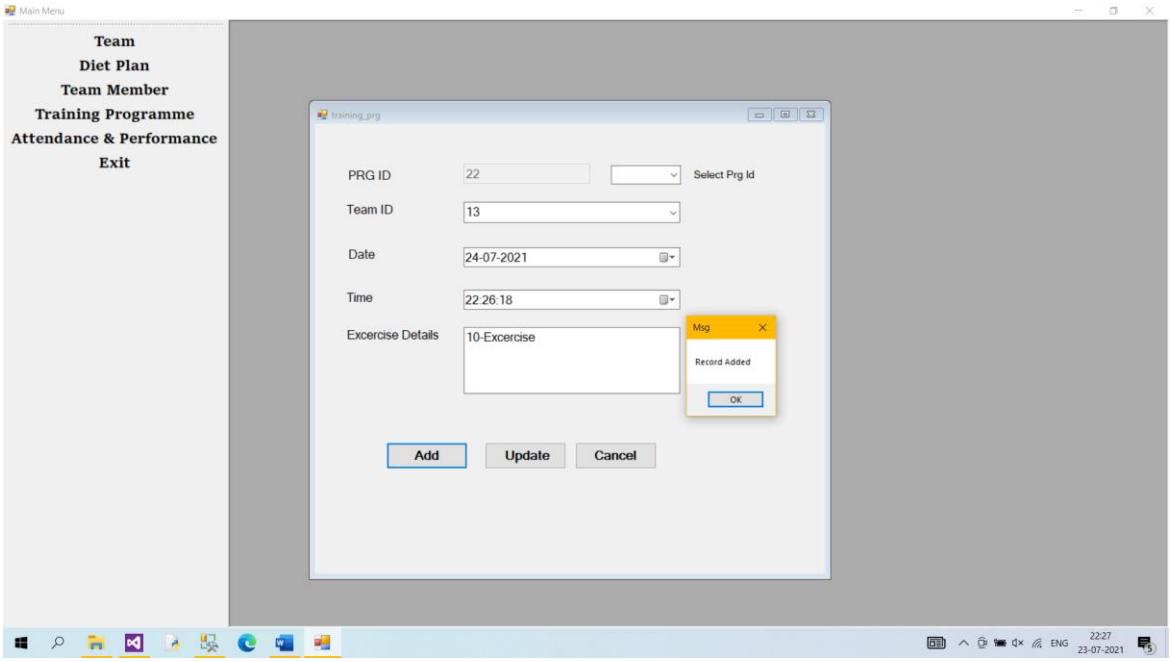

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team_Member
Team Member Add/ Search/Update/Delete
Mem ID
102
Select Mem Id to Search
Team Id
13
Member's Name
Subodh kr
Email
skr@gmail.com
Address
Gurugram, Haryana
Contact
9876765654
Age
14
Height(in Cm)
156
Weight (in Kg)
45
Role
Batman
Add
Update
Cancel
Delete
Msg
Record Added
OK
ENG
22:25
23-07-2021

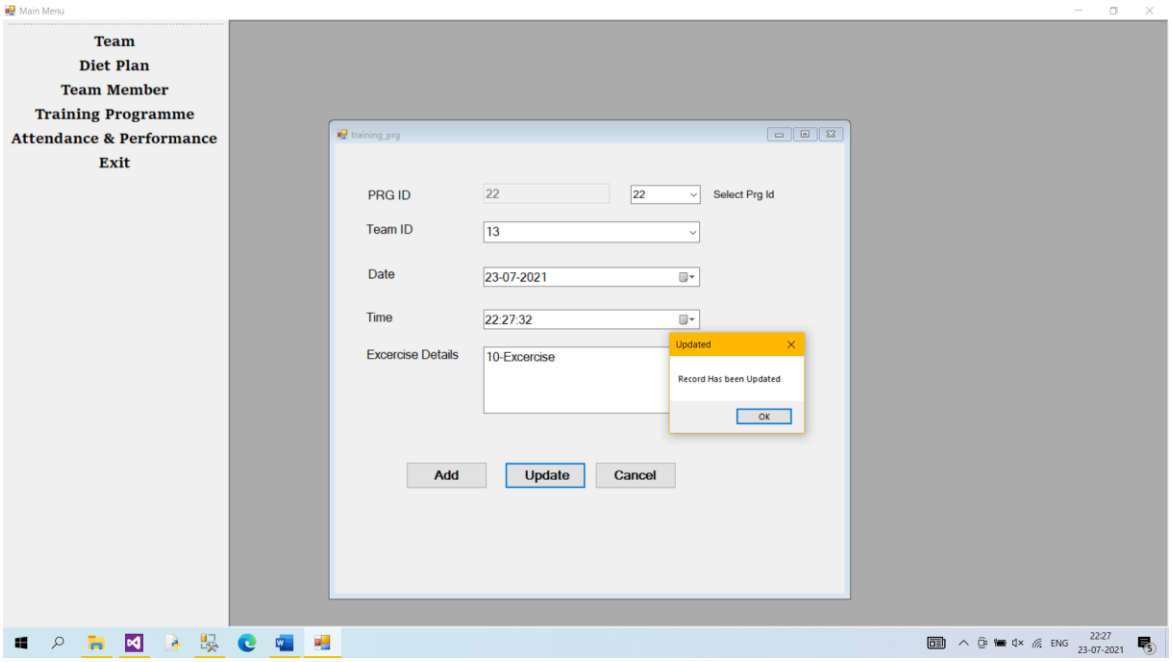

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team_Member
Team Member Add/ Search/Update/Delete
Mem ID
103
Select Mem Id to Search
101
102
Team Id
Member's Name
Email
Address
Contact
Age
Height(in Cm)
Weight (in Kg)
Role
Add
Update
Cancel
Delete
ENG
22:25
23-07-2021

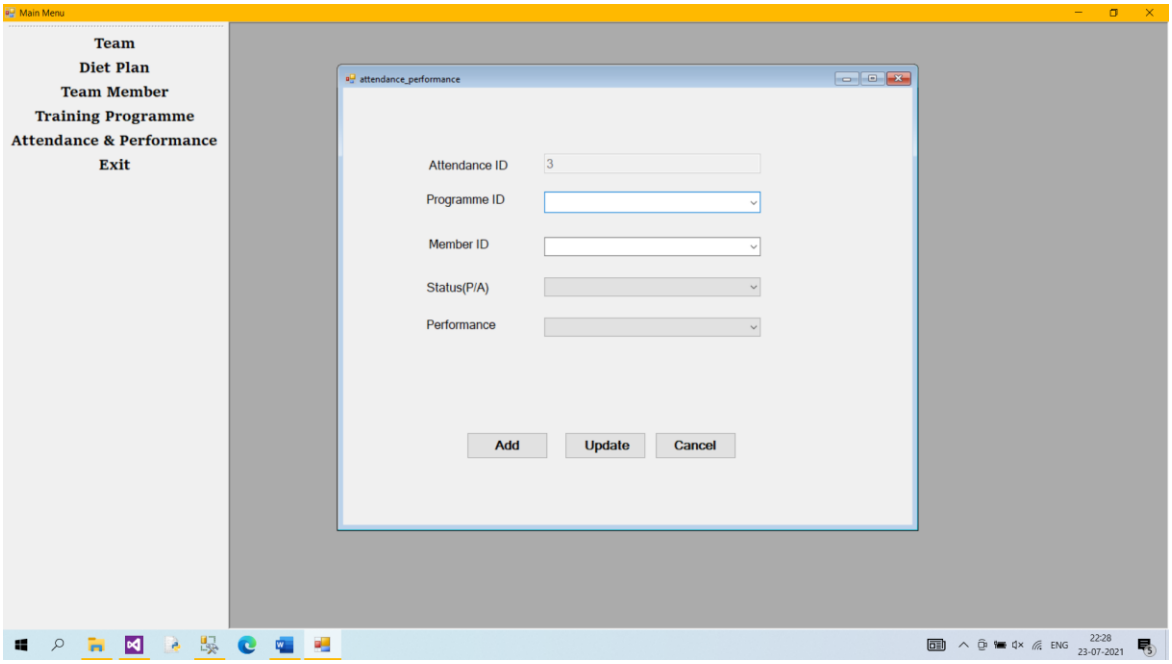

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
Team_Member
Team Member Add/ Search/Update/Delete
Mem ID
102
102
Select Mem Id to Search
Team Id
13
Member's Name
Subodh kr
Email
skr@gmail.com
Address
Gurugram, Haryana
Contact
9876765654
Age
13
Height(in Cm)
156
Weight (in Kg)
45
Role
Batman
Updated
Record Has been Updated
OK
Add
Update
Cancel
Delete
ENG
22:26
23-07-2021

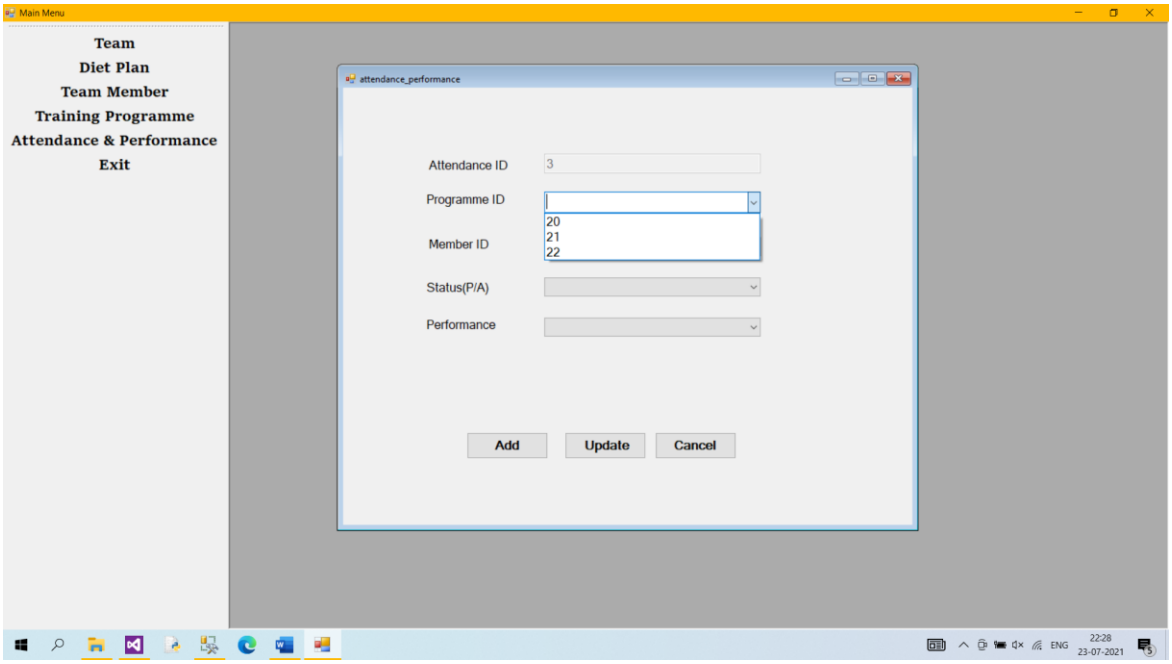

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
training_prg
PRG ID
22
Select Prg Id
Team ID
Date
23-07-2021
Time
22:26:18
Excercise Details
Add
Update
Cancel
ENG
22:26
23-07-2021

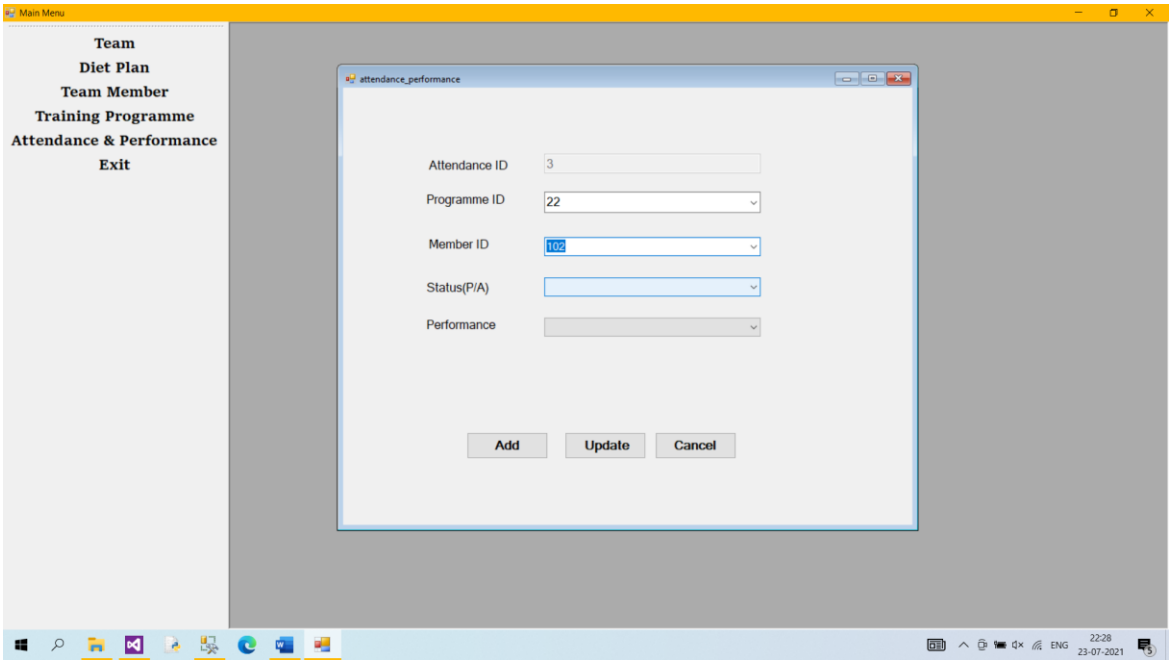

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
training_prg
PRG ID
22
Select Prg Id
Team ID
13
Date
24-07-2021
Time
22:26:18
Excercise Details
10-Excercise
Add
Update
Cancel
ENG
22:27
23-07-2021

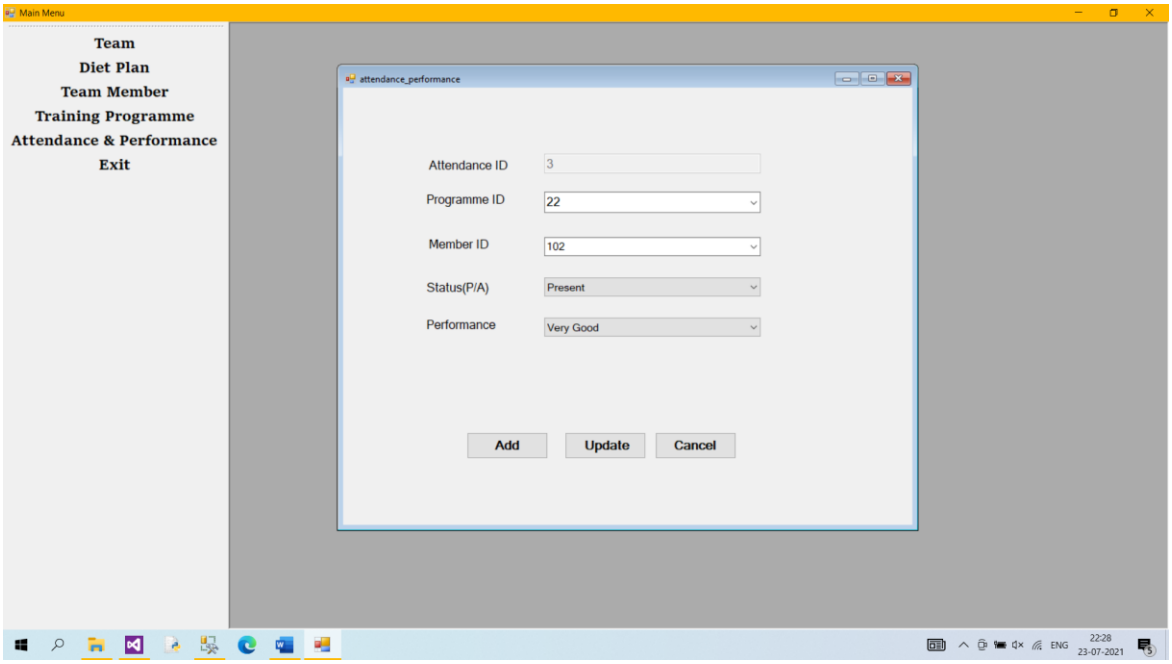

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
training_prg
PRG ID
22
Select Prg Id
Team ID
13
Date
24-07-2021
Time
22:26:18
Excercise Details
10-Excercise
Msg
Record Added
OK
Add
Update
Cancel
ENG
22:27
23-07-2021

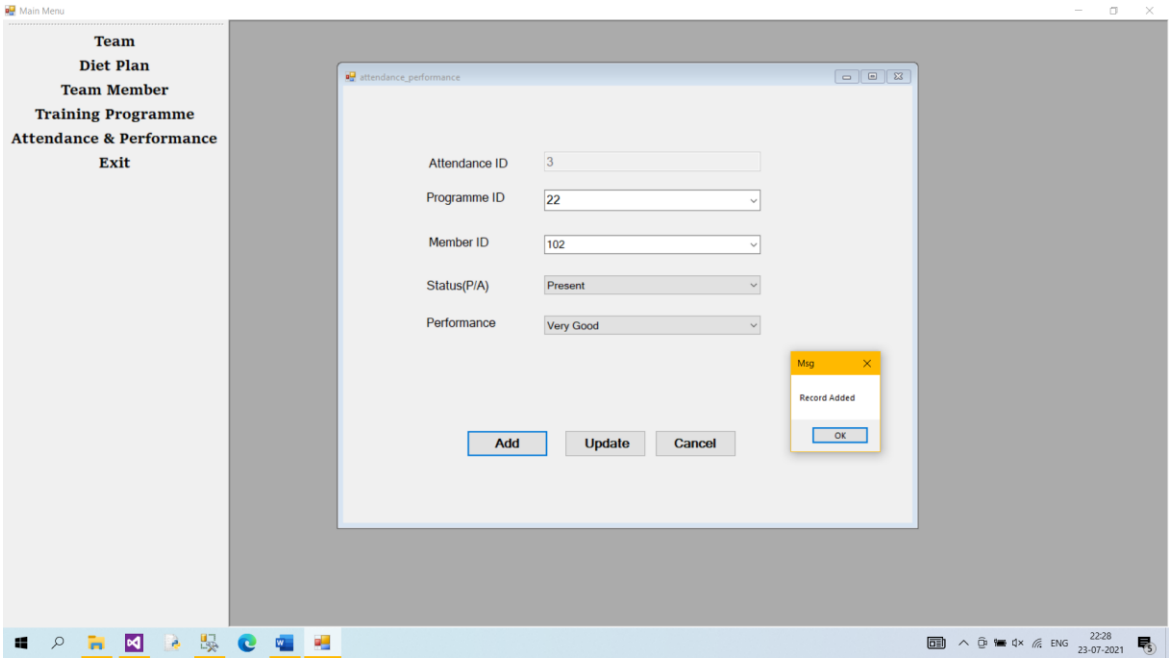

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
training_prg
PRG ID
22
22
Select Prg Id
Team ID
13
Date
23-07-2021
Time
22:27:32
Excercise Details
10-Excercise
Updated
Record Has been Updated
OK
Add
Update
Cancel
ENG
22:27
23-07-2021


Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
attendance_performance
Attendance ID
3
Programme ID
Member ID
Status(P/A)
Performance
Add
Update
Cancel
ENG
22:28
23-07-2021

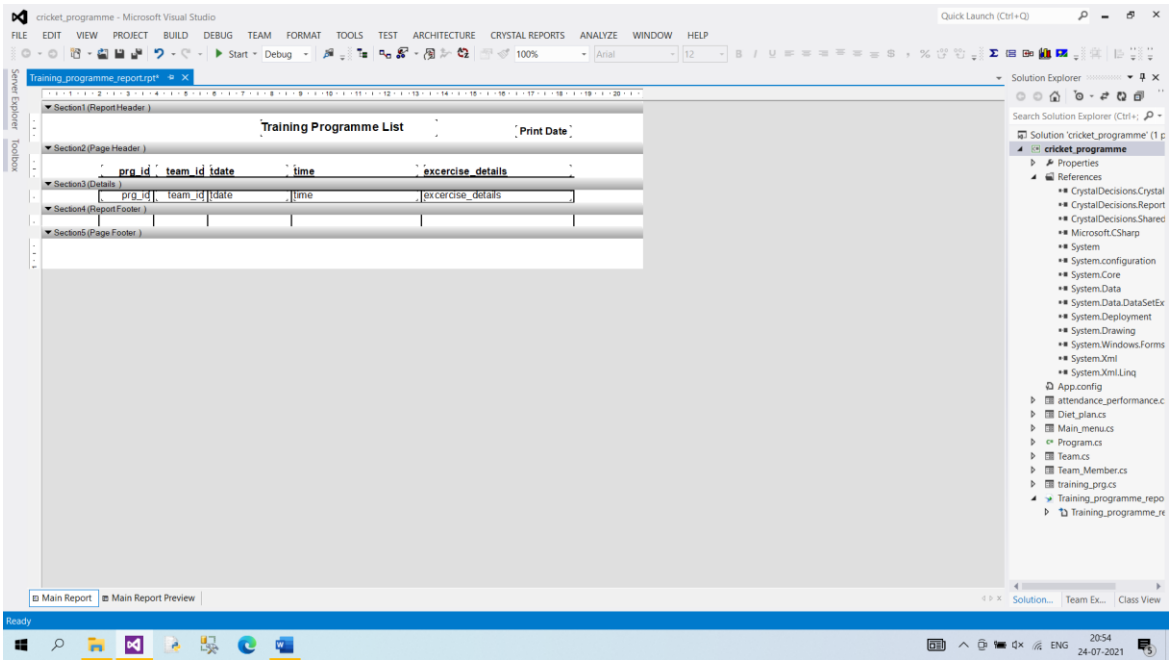

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
attendance_performance
Attendance ID
3
Programme ID
20
21
22
Member ID
Status(P/A)
Performance
Add
Update
Cancel
ENG
22:28
23-07-2021

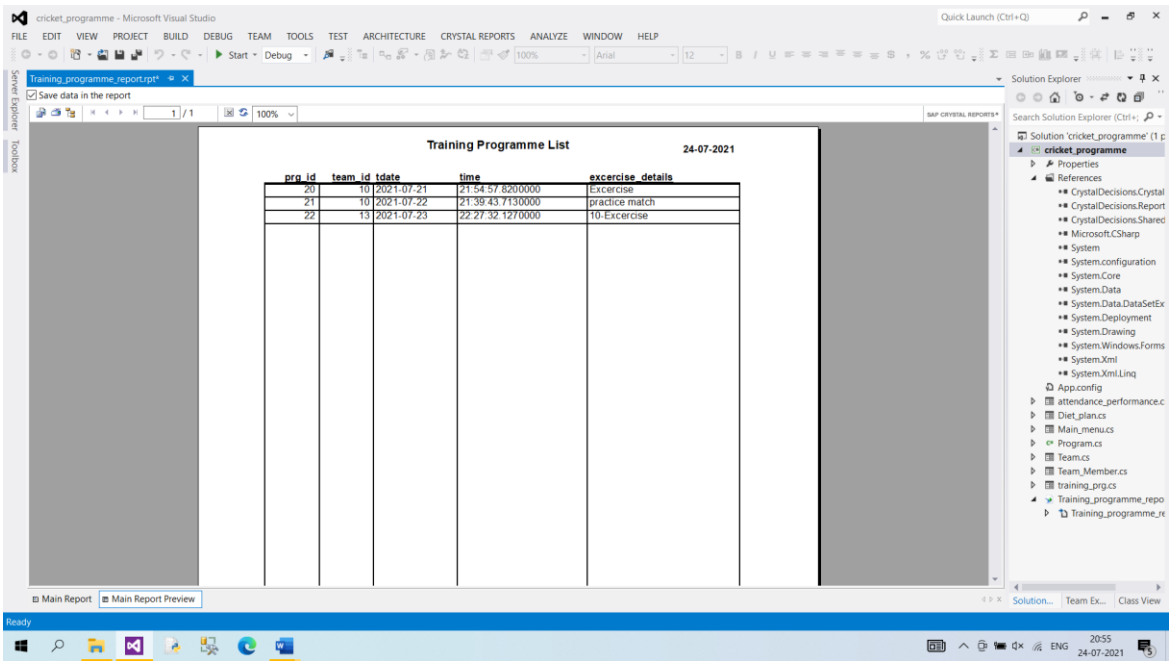

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
attendance_performance
Attendance ID
3
Programme ID
22
Member ID
102
Status(P/A)
Performance
Add
Update
Cancel
ENG
22:28
23-07-2021

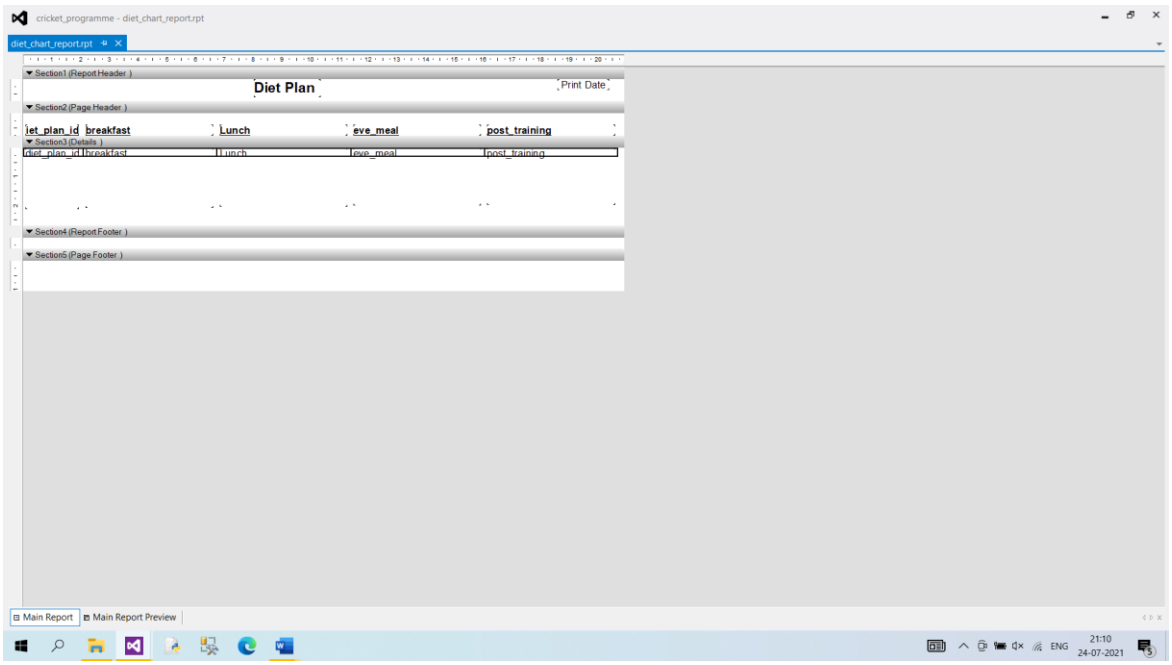

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
attendance_performance
Attendance ID
3
Programme ID
22
Member ID
102
Status(P/A)
Present
Performance
Very Good
Add
Update
Cancel
ENG
22:28
23-07-2021

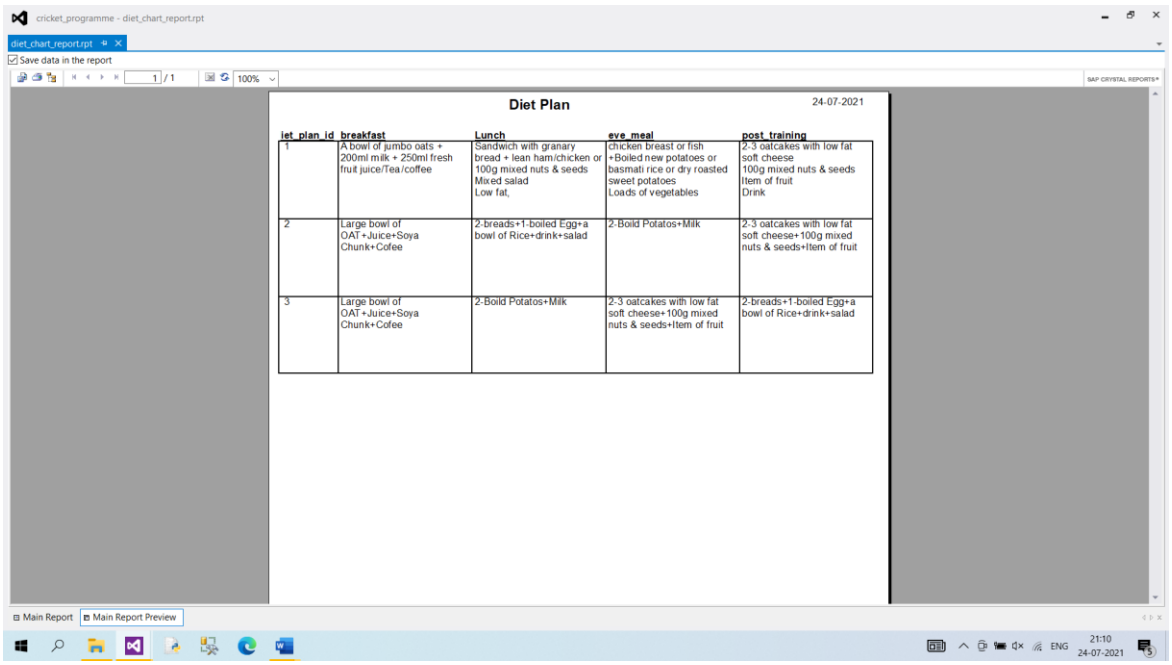

Main Menu
Team
Diet Plan
Team Member
Training Programme
Attendance & Performance
Exit
attendance_performance
Attendance ID
3
Programme ID
22
Member ID
102
Status(P/A)
Present
Performance
Very Good
Add
Update
Cancel
Msg
Record Added
OK
ENG
22:28
23-07-2021

Report


cricket_programme - Microsoft Visual Studio
Training_programme_report.rpt*
Section1 (Report Header)
Training Programme List
Print Date
Section2 (Page Header)
prg_id
team_id
tdate
time
excercise_details
Section3 (Details)
Section4 (Report Footer)
Section5 (Page Footer)
Main Report
Main Report Preview
Solution Explorer
cricket_programme
References
Ready


cricket_programme - Microsoft Visual Studio
Training_programme_report.rpt*
Save data in the report
Training Programme List
24-07-2021
prg_id team_id tdate time excercise_details
20 10 2021-07-21 21:54:57.8200000 Excercise
21 10 2021-07-22 21:39:43.7130000 practice match
22 13 2021-07-23 22:27:32.1270000 10-Excercise
Main Report
Main Report Preview
Solution Explorer
cricket_programme
Ready


cricket_programme - diet_chart_report.rpt
diet_chart_report.rpt
Section1 (Report Header)
Diet Plan
Print Date
Section2 (Page Header)
let_plan_id
breakfast
Lunch
eve_meal
post_training
Section3 (Details)
diet_plan_id
breakfast
Lunch
eve_meal
post_training
Section4 (Report Footer)
Section5 (Page Footer)
Main Report
Main Report Preview
ENG
21:10
24-07-2021


cricket_programme - diet_chart_report.rpt
diet_chart_report.rpt
Save data in the report
Diet Plan
24-07-2021
jet_plan_id breakfast Lunch eve_meal post_training
1 A bowl of jumbo oats + 200ml milk + 250ml fresh fruit juice/Tea/coffee Sandwich with granary bread + lean ham/chicken or 100g mixed nuts & seeds Mixed salad Low fat, chicken breast or fish +Boiled new potatoes or basmati rice or dry roasted sweet potatoes Loads of vegetables 2-3 oatcakes with low fat soft cheese 100g mixed nuts & seeds Item of fruit Drink
2 Large bowl of OAT+Juice+Soya Chunk+Cofee 2-breads+1-boiled Egg+a bowl of Rice+drink+salad 2-Boild Potatos+Milk 2-3 oatcakes with low fat soft cheese+100g mixed nuts & seeds+Item of fruit
3 Large bowl of OAT+Juice+Soya Chunk+Cofee 2-Boild Potatos+Milk 2-3 oatcakes with low fat soft cheese+100g mixed nuts & seeds+Item of fruit 2-breads+1-boiled Egg+a bowl of Rice+drink+salad
Main Report
Main Report Preview
ENG
21:10
24-07-2021

4.5 SECURITY Issues

Security is the most important part of any system. It can be either the security of system program functionalities or underlying database. We have very cautious process of authentication of user that no one could change its contents in unauthorized manner.

Security and integrity of database are very important for any software system because databases are the backbone of the system. Security needs to be implemented at every level of the system so that only authorized user can access the system for updating and other significance process.

Entering correct password while opening the system or we can say that entering the system is the process of authentication. If anyone is entering the password is wrong then he/she cannot access the system for any change purpose.

The main purpose of the security is to save system from accidentally changes or loss of information or also getting wrong information. The system administrator is the person that can change the information or update the information. He can also grant the permission that who has to enter the system and what can he do. So, security is the most important topic to be concerned.

4.6 Test Case Design

The contents required to test a program is included in test cases ultimately; testing comes down to selecting and executing essential pieces of information.

- A set of inputs.
- The expected results when the inputs are expected.
- The conditions for execution of code.

The component of test cases can be different or they will be behaved differently in web-based application from the component of test cases in a arithmetical project.

For instance, taking the example of employee entry form of my project is generating exception by giving wrong input to it (the concept of robustness).

In a desktop-based application, we can handle the exceptions (Run-time errors) generated by inputting wrong value in two ways.

1. By putting the code in try block to throw the exception.

2. By using client-side validation. So, in the client machine so that the user could not enter wrong input and he could know that which type of value should be input.

Test Cases

Team member

Case-1: leave some blank fields

Expected Result- Error Message please fill entire field

Output –please fill the entire field

Result- Pass

Case-2: Input Number in Name Field

Expected Result: Enter character only.

Output- please enter character only

Result –pass

Case -3: Input correct data in fields.

Expected Result: Record will be saved.

Output: Saved

Result- Pass

All modules are tested in same way.

Chapter-5

TESTING AND IMPLEMENTATION

5.1 Implementation Approaches

1) Parallel System

In this method, both systems are running in parallel over an introductory period. The old system remains operational while the new system comes on line. The advantages are that the organization is left with a full fallback position in the event of new system failure, processing errors or processes having been omitted. It safeguards against any loss of revenue, service to customers or loss of time in the event of system failure. It is more secure method.

2) Direct Cutover

In this method, new system directly replaces an old system if system is fully reliable. The benefits of using the new system are clearly immediate and it requires meticulous planning in terms of both people and system being complete and prepared at the key moment in time. It has high risks; if system has no reliability, then organization may face losses.

3) Pilot Approach

The group that uses the new system first is called the pilot site. The old system continues to operate for the entire organization including the pilot site. After the system proves successful at the pilot site, it is implemented in the rest of the organization,

This is *less expensive* than the parallel operation as only at one section both system works for limited period.

4) Phase-In Method

Phased operation works in different phases or stages. In this method, Implementation of new system in modules or stages is called phased operation. In phased adoption implementation, the organization is gradually transferring to a new system in different phases as per module or sub-system. A part of system is provided to the users.

5.2 Coding Details and Code Efficiency

5.2.1 Code Efficiency

Interface Module

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace cricket_programme
{
   public partial class Main_menu : Form
   {
      public Main_menu()
      {
         InitializeComponent();
      }

      private void teamToolStripMenuItem_Click(object sender, EventArgs e)
      {
         Team tt = new Team();
         tt.Show();
         tt.MdiParent = this;
      }
```

```
        private void dietPlanToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Diet_plan dp = new Diet_plan();
            dp.Show();
            dp.MdiParent = this;
        }

        private void teamMemberToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Team_Member tm = new Team_Member();
            tm.Show();
            tm.MdiParent = this;
        }

        private void trainingProgrammeToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            training_prg tp = new training_prg();
            tp.Show();
            tp.MdiParent = this;
        }

        private void attendancePerformanceToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            attendance_performance ap = new attendance_performance();
            ap.Show();
```

```
            ap.MdiParent = this;
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void Main_menu_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Team Module

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Configuration;
using System.Data.SqlClient;
namespace cricket_programme
```

```
{
    public partial class Team : Form
    {
        SqlConnection con1 = new
SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].Connecti
onString); // This is connection String

        public Team()
        {
            InitializeComponent();
        }

        private void Team_Load(object sender, EventArgs e)
        {
            int a = 0;
            con1.Open();
            SqlCommand cmd2 = new SqlCommand("Select max(team_id) from
team", con1); // Select query

            SqlDataReader dr2 = cmd2.ExecuteReader();

            while (dr2.Read())
            {
                a = Int32.Parse(dr2[0].ToString());

            }
            a = a + 1;
            textBox1.Text = a.ToString();
            con1.Close();
```

```
        {
            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from
diet_plan", con1); // Select query
            SqlDataReader dr1 = cmd1.ExecuteReader();

            while (dr1.Read())
            {
                comboBox1.Items.Add(dr1[0].ToString());

            }
            con1.Close();

        }
        {
            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from team",
con1); // Select query
            SqlDataReader dr1 = cmd1.ExecuteReader();

            while (dr1.Read())
            {
                comboBox2.Items.Add(dr1[0].ToString());

            }
            con1.Close();

        }
```

```
        }
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {

            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from diet_plan
where diet_plan_id=@a", con1); // Select query
            cmd1.Parameters.AddWithValue("@a", comboBox1.Text);
            SqlDataReader dr1 = cmd1.ExecuteReader();
            while (dr1.Read())
            {
                textBox10.Text = dr1[0].ToString();
                textBox9.Text = dr1[1].ToString();
                textBox8.Text = dr1[2].ToString();
                textBox7.Text = dr1[3].ToString();
                textBox6.Text = dr1[4].ToString();
            }
            con1.Close();
        }

        private void comboBox2_SelectedIndexChanged(object sender,
EventArgs e)
        {

            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from team where
team_id=@a", con1); // Select query
            cmd1.Parameters.AddWithValue("@a", comboBox2.Text);
            SqlDataReader dr1 = cmd1.ExecuteReader();
```

```
            while (dr1.Read())
            {
                textBox1.Text = dr1[0].ToString();
                textBox2.Text = dr1[1].ToString();
                textBox3.Text = dr1[2].ToString();
                textBox4.Text = dr1[3].ToString();
               //comboBox1.Text = dr1[4].ToString();
               textBox5.Text = dr1[5].ToString();
            }
            con1.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == ""
|| textBox4.Text == "" || textBox5.Text == "")
                MessageBox.Show("Fill the data", "Msg");
            else
            {
                try
                {
                    con1.Open();
                    SqlCommand cmd = new SqlCommand("insert into team
values(@a,@b,@c,@d,@e,@f)", con1); // Select query
                    cmd.Parameters.AddWithValue("@a", textBox1.Text);
                    cmd.Parameters.AddWithValue("@b", textBox2.Text);
                    cmd.Parameters.AddWithValue("@c", textBox3.Text);
                    cmd.Parameters.AddWithValue("@d", textBox4.Text);
                    cmd.Parameters.AddWithValue("@e", comboBox1.Text);
```

```
            cmd.Parameters.AddWithValue("@f", textBox5.Text);

            SqlDataReader dr = cmd.ExecuteReader();
            MessageBox.Show("Record Added", "Msg");
            con1.Close();
            this.Close();
        }
        catch (Exception ee)
        {
            MessageBox.Show("Exception Error" + ee.ToString());
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    con1.Open();
    SqlCommand cmd = new SqlCommand("update team set
team_name=@a, age_group=@b, description=@c, diet_plan_id=@d,
experience=@e where diet_plan_id=@k", con1); // Update query
    cmd.Parameters.AddWithValue("@k", comboBox1.Text);

    cmd.Parameters.AddWithValue("@a", textBox2.Text);
    cmd.Parameters.AddWithValue("@b", textBox3.Text);
    cmd.Parameters.AddWithValue("@c", textBox4.Text);
    cmd.Parameters.AddWithValue("@d", comboBox1.Text);
    cmd.Parameters.AddWithValue("@e", textBox5.Text);
    SqlDataReader dr = cmd.ExecuteReader();
```

```
            MessageBox.Show("Record Has been Updated", "Updated");
            con1.Close();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Team member Module

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Configuration;
using System.Data.SqlClient;

namespace cricket_programme
{
    public partial class Team_Member : Form
    {
```

```
SqlConnection con1 = new SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].ConnectionString); // This is connection String

public Team_Member()
{
   InitializeComponent();
}

private void Team_Member_Load(object sender, EventArgs e)
{
   int a = 0;
   con1.Open();
   SqlCommand cmd2 = new SqlCommand("Select max(mem_id) from team_member", con1); // Select query

   SqlDataReader dr2 = cmd2.ExecuteReader();

   while (dr2.Read())
   {
      a = Int32.Parse(dr2[0].ToString());

   }
   a = a + 1;
   textBox1.Text = a.ToString();
   con1.Close();
   {
      con1.Open();
```

```
            SqlCommand cmd1 = new SqlCommand("Select * from team",
con1); // Select query
            SqlDataReader dr1 = cmd1.ExecuteReader();

            while (dr1.Read())
            {
               comboBox1.Items.Add(dr1[0].ToString());

            }
            con1.Close();
            {
               con1.Open();
               SqlCommand cmd3 = new SqlCommand("Select * from
team_member", con1); // Select query
               SqlDataReader dr3 = cmd3.ExecuteReader();

               while (dr3.Read())
               {
                  comboBox2.Items.Add(dr3[0].ToString());

               }
               con1.Close();
            }
         }
      }

      private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
      {
```

```
    }

    private void comboBox2_SelectedIndexChanged(object sender,
EventArgs e)
    {
        con1.Open();
        SqlCommand cmd1 = new SqlCommand("Select * from
team_member where mem_id=@a", con1); // Select query
        cmd1.Parameters.AddWithValue("@a", comboBox2.Text);
        SqlDataReader dr1 = cmd1.ExecuteReader();
        while (dr1.Read())
        {
            textBox1.Text = dr1[0].ToString();
            comboBox1.Text = dr1[1].ToString();
            textBox3.Text = dr1[2].ToString();
            textBox4.Text = dr1[3].ToString();

            textBox5.Text = dr1[4].ToString();
            textBox6.Text = dr1[5].ToString();

            textBox7.Text = dr1[6].ToString();
            textBox8.Text = dr1[7].ToString();
            textBox9.Text = dr1[8].ToString();
            textBox10.Text = dr1[9].ToString();
        }
        con1.Close();
    }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text == "" || textBox4.Text == "" || textBox3.Text == ""
|| textBox4.Text == "" || textBox5.Text == "" || textBox5.Text == "")
                MessageBox.Show("Fill the data", "Msg");
            else
            {
                try
                {

                    con1.Open();
                    SqlCommand cmd = new SqlCommand("insert into
team_member values(@a,@b,@c,@d,@e,@f,@g,@h,@i,@j)", con1); //
insert query
                    cmd.Parameters.AddWithValue("@a", textBox1.Text);
                    cmd.Parameters.AddWithValue("@b", comboBox1.Text);
                    cmd.Parameters.AddWithValue("@c", textBox3.Text);
                    cmd.Parameters.AddWithValue("@d", textBox4.Text);
                    cmd.Parameters.AddWithValue("@e", textBox5.Text);
                    cmd.Parameters.AddWithValue("@f", textBox6.Text);
                    cmd.Parameters.AddWithValue("@g", textBox7.Text);
                    cmd.Parameters.AddWithValue("@h", textBox8.Text);
                    cmd.Parameters.AddWithValue("@i", textBox9.Text);
                    cmd.Parameters.AddWithValue("@j", textBox10.Text);

                    SqlDataReader dr = cmd.ExecuteReader();
                    MessageBox.Show("Record Added", "Msg");
                    con1.Close();
                }
```

```
            catch (Exception ee)
            {
                MessageBox.Show("Exception Error" + ee.ToString());
            }
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        con1.Open();
        SqlCommand cmd = new SqlCommand("update team_member set
team_id=@a, mname=@b, email=@c, address=@d, contact=@e,
age=@f,height_cm=@g,weight_kg=@h,role=@i  where mem_id=@k",
con1); // Update query
        cmd.Parameters.AddWithValue("@k", comboBox2.Text);

        cmd.Parameters.AddWithValue("@a", comboBox1.Text);
        cmd.Parameters.AddWithValue("@b", textBox3.Text);
        cmd.Parameters.AddWithValue("@c", textBox4.Text);
        cmd.Parameters.AddWithValue("@d", textBox5.Text);

        cmd.Parameters.AddWithValue("@e", textBox6.Text);
        cmd.Parameters.AddWithValue("@f", textBox7.Text);
        cmd.Parameters.AddWithValue("@g", textBox8.Text);

        cmd.Parameters.AddWithValue("@h", textBox9.Text);
        cmd.Parameters.AddWithValue("@i", textBox10.Text);
```

```
            SqlDataReader dr = cmd.ExecuteReader();

            MessageBox.Show("Record Has been Updated", "Updated");
            con1.Close();
        }
        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void button4_Click(object sender, EventArgs e)
        {
            try
            {
                con1.Open();
                SqlCommand cmd = new SqlCommand("delete from
team_member where mem_id=@k", con1); // Delete query
                cmd.Parameters.AddWithValue("@k", comboBox2.Text);
                SqlDataReader dr = cmd.ExecuteReader();
                MessageBox.Show("Record Has been Deleted", "Deleted");
                con1.Close();
            }
            catch (Exception ee)
            {
            MessageBox.Show("Exception....."+ee.ToString());
            }
        }
    }
}
```

Training programme Module

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Configuration;
using System.Data.SqlClient;

namespace cricket_programme
{
   public partial class training_prg : Form
   {
      SqlConnection con1 = new
SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].Connecti
onString); // This is connection Stri
      public training_prg()
      {
         InitializeComponent();
      }

      private void training_prg_Load(object sender, EventArgs e)
      {
         int a = 0;
         con1.Open();
```

```
        SqlCommand cmd2 = new SqlCommand("Select max(prg_id) from
training_programme", con1); // Select query

        SqlDataReader dr2 = cmd2.ExecuteReader();

        while (dr2.Read())
        {
            a = Int32.Parse(dr2[0].ToString());

        }
        a = a + 1;
        textBox1.Text = a.ToString();
        con1.Close();
        {
            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from team",
con1); // Select query
            SqlDataReader dr1 = cmd1.ExecuteReader();

            while (dr1.Read())
            {
                comboBox1.Items.Add(dr1[0].ToString());

            }
            con1.Close();
        }
        con1.Close();
        {
            con1.Open();
```

```
            SqlCommand cmd1 = new SqlCommand("Select * from
training_programme", con1); // Select query
            SqlDataReader dr1 = cmd1.ExecuteReader();

            while (dr1.Read())
            {
               comboBox2.Items.Add(dr1[0].ToString());

            }
            con1.Close();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {

            con1.Open();
            SqlCommand cmd = new SqlCommand("insert into
training_programme values(@a,@b,@c,@d,@e)", con1); // insert query
            cmd.Parameters.AddWithValue("@a", textBox1.Text);
            cmd.Parameters.AddWithValue("@b", comboBox1.Text);
            cmd.Parameters.AddWithValue("@c", dateTimePicker1.Value);
            cmd.Parameters.AddWithValue("@d", dateTimePicker2.Value);
            cmd.Parameters.AddWithValue("@e", textBox4.Text);
            SqlDataReader dr = cmd.ExecuteReader();
            MessageBox.Show("Record Added", "Msg");
            con1.Close();
```

```
            }
            catch (Exception ee)
            {
                MessageBox.Show("Exception Error" + ee.ToString());
            }
            this.Close();
        }

        private void comboBox2_SelectedIndexChanged(object sender,
EventArgs e)
        {
            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from
training_programme where prg_id=@a", con1); // Select query
            cmd1.Parameters.AddWithValue("@a", comboBox2.Text);
            SqlDataReader dr1 = cmd1.ExecuteReader();
            while (dr1.Read())
            {
                textBox1.Text = dr1[0].ToString();
                comboBox1.Text = dr1[1].ToString();

                textBox4.Text = dr1[4].ToString();
            }
            con1.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            con1.Open();
```

```
            SqlCommand cmd = new SqlCommand("update training_programme
set team_id=@a, tdate=@b, time=@c, excercise_details=@d where
prg_id=@k", con1); // Update query
            cmd.Parameters.AddWithValue("@k", comboBox2.Text);

            cmd.Parameters.AddWithValue("@a", comboBox1.Text);
            cmd.Parameters.AddWithValue("@b", dateTimePicker1.Value);
            cmd.Parameters.AddWithValue("@c", dateTimePicker1.Value);
            cmd.Parameters.AddWithValue("@d", textBox4.Text);
            SqlDataReader dr = cmd.ExecuteReader();

            MessageBox.Show("Record Has been Updated", "Updated");
            con1.Close();
            this.Close();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Diet Plan

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Configuration;
using System.Data.SqlClient;
namespace cricket_programme
{
   public partial class Diet_plan : Form
   {
      SqlConnection con1 = new
SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].Connecti
onString); // This is connection String
      public Diet_plan()
      {
         InitializeComponent();
      }

      private void Form1_Load(object sender, EventArgs e)
      {
         int a = 0;
         con1.Open();
```

```
        SqlCommand cmd2 = new SqlCommand("Select max(diet_plan_id) from diet_plan", con1); // Select query

        SqlDataReader dr2 = cmd2.ExecuteReader();

        while (dr2.Read())
        {
            a = Int32.Parse(dr2[0].ToString());

        }
        a = a + 1;
        textBox1.Text = a.ToString();
        con1.Close();
        {
            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from diet_plan", con1); // Select query
            SqlDataReader dr1 = cmd1.ExecuteReader();

            while (dr1.Read())
            {
                comboBox1.Items.Add(dr1[0].ToString());

            }
            con1.Close();
        }

    }
```

```
        private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            con1.Open();
            SqlCommand cmd1 = new SqlCommand("Select * from diet_plan
where diet_plan_id=@a", con1); // Select query
            cmd1.Parameters.AddWithValue("@a", comboBox1.Text);
            SqlDataReader dr1 = cmd1.ExecuteReader();
            while (dr1.Read())
            {
                textBox1.Text = dr1[0].ToString();
                textBox2.Text = dr1[1].ToString();
                textBox3.Text = dr1[2].ToString();
                textBox4.Text = dr1[3].ToString();
                textBox5.Text = dr1[4].ToString();
            }
            con1.Close();
        }

        private void button1_Click(object sender, EventArgs e)
        {

            con1.Close();
            if (textBox1.Text == "" || textBox2.Text == "" || textBox3.Text == ""
|| textBox4.Text == "" || textBox5.Text == "")
                MessageBox.Show("Fill the data", "Msg");
            {
                try
                {
```

```
            con1.Open();
            SqlCommand cmd = new SqlCommand("insert into diet_plan
values(@a,@b,@c,@d,@e)", con1); // insert query
            cmd.Parameters.AddWithValue("@a", textBox1.Text);
            cmd.Parameters.AddWithValue("@b", textBox2.Text);
            cmd.Parameters.AddWithValue("@c", textBox3.Text);
            cmd.Parameters.AddWithValue("@d", textBox4.Text);
            cmd.Parameters.AddWithValue("@e", textBox5.Text);



            SqlDataReader dr = cmd.ExecuteReader();
            MessageBox.Show("Record Added", "Msg");
            con1.Close();
        }
        catch (Exception ee)
        {
            MessageBox.Show("Exception Error" + ee.ToString());
        }
    }

}

private void button3_Click(object sender, EventArgs e)
{
    con1.Open();
    SqlCommand cmd = new SqlCommand("update diet_plan set
breakfast=@a, Lunch=@b, eve_meal=@c, post_training=@d where
diet_plan_id=@k", con1); // update query
```

```
            cmd.Parameters.AddWithValue("@k", comboBox1.Text);

            cmd.Parameters.AddWithValue("@a", textBox2.Text);
            cmd.Parameters.AddWithValue("@b", textBox3.Text);
            cmd.Parameters.AddWithValue("@c", textBox4.Text);
            cmd.Parameters.AddWithValue("@d", textBox5.Text);
            SqlDataReader dr = cmd.ExecuteReader();

            MessageBox.Show("Record Has been Updated", "Updated");
            con1.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Attendance & performance

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Configuration;
```

```
using System.Data.SqlClient;
namespace cricket_programme
{
   public partial class attendance_performance : Form
   {
      SqlConnection con1 = new
SqlConnection(ConfigurationManager.ConnectionStrings["mycon"].Connecti
onString); // This is connection Stri

      public attendance_performance()
      {
         InitializeComponent();
      }

      private void attendance_performance_Load(object sender, EventArgs e)
      {
         {
            int a = 0;
            con1.Open();
            SqlCommand cmd2 = new SqlCommand("Select max(attd_id)
from attendance_performance", con1); // Select query

            SqlDataReader dr2 = cmd2.ExecuteReader();

            while (dr2.Read())
            {
               a = Int32.Parse(dr2[0].ToString());
            }
            a = a + 1;
```

```
            textBox1.Text = a.ToString();
            con1.Close();
        }
        con1.Open();
        SqlCommand cmd1 = new SqlCommand("Select * from
training_programme", con1); // Select query
        SqlDataReader dr1 = cmd1.ExecuteReader();

        while (dr1.Read())
        {
            comboBox1.Items.Add(dr1[0].ToString());
        }
        con1.Close();
        {
            con1.Open();
            SqlCommand cmd2 = new SqlCommand("Select * from
team_member", con1); // Select query
            SqlDataReader dr2 = cmd2.ExecuteReader();
            while (dr2.Read())
            {
                comboBox2.Items.Add(dr2[0].ToString());
            }
            con1.Close();
        }

    }

    private void button1_Click(object sender, EventArgs e)
    {
```

```
            try
            {

                con1.Open();
                SqlCommand cmd = new SqlCommand("insert into
attendance_performance values(@a,@b,@c,@d,@e)", con1); // Select query
                cmd.Parameters.AddWithValue("@a", textBox1.Text);
                cmd.Parameters.AddWithValue("@b", comboBox1.Text);
                cmd.Parameters.AddWithValue("@c", comboBox2.Text);
                cmd.Parameters.AddWithValue("@d", comboBox3.Text);
                cmd.Parameters.AddWithValue("@e", comboBox4.Text);
                SqlDataReader dr = cmd.ExecuteReader();
                MessageBox.Show("Record Added", "Msg");
                con1.Close();
            }
            catch (Exception ee)
            {
                MessageBox.Show("Exception Error" + ee.ToString());
            }
            this.Close();
        }
        private void button3_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

5.3 Testing Approaches

5.3.1 Unit Testing

The purpose of this phase is to test the individual units of the developing software component. This phase is recursive and is to be repeated, as many as there are, levels of testing. In the DGLW project, each individual form has been tested using techniques of testing namely: Client-side testing using JavaScript.

Each individual form has been validated so that user enters only valid data at every time

5.3.2 Integrated Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

5.4 Modification and Improvement

The system is still under construction few reports are yet to me made after that this system will be implanted at client side. Users will be given a training to use the package and special workshops are conducted for the purpose. And according to their feedback the changes are implanted in the software.

CHAPTER-6

RESULT & DISCUSSION

- *Test Report*
- *User Documentation*

6.1 Test Report

Testing Laws –

There are rules (laws) to be followed while performing the testing process. These are as follows.

1. Testing can only be used to show the presence of error not absence of error.
2. A combination of different verification and validation method out perform a single method alone.
3. Developers are unsuited to test their own code.
4. Approximately 80% of errors are found in 20% of code.
5. Partition testing, which is new test criteria is well suited for long project testing.
6. The adequacy of a test suite for the coverage criterion can only be defined intuitively.

This test plan relates to the following System (i.e. object under test):

System Name	Cricket training management
System Version	101
Test Run	3

The tested version is labeled **xxxx**.

Acceptance Criteria

In general, a test case is "PASSED" when the real outcome corresponds to the expected outcome, otherwise the test case is marked as “FAILED”. Expected outcomes are defined separately in each test case.

- a 100% test coverage is required (i.e. all implemented changes need to be tested at least once)
- no critical defects are tolerated

6.2 User Documentation

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the **Cricket Training Management System.** It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate.

1.2. Scope of Project

This software system will be a **Cricket Training Management System** for local users of an organization. This system will be designed to maximize the organization's productivity by providing tools to assist in automating process,

1.3. Glossary

Term	Definition
Database	Collection of all the information monitored by this system.
User	Person who interacts sudents and system functions. Employee of organizations.
Field	A cell within a form.
Historical Society Database	The existing membership database.
Member	A member of the Historical Society listed in the HS database.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Admin	Reviewer or owner of organization

1.4. References

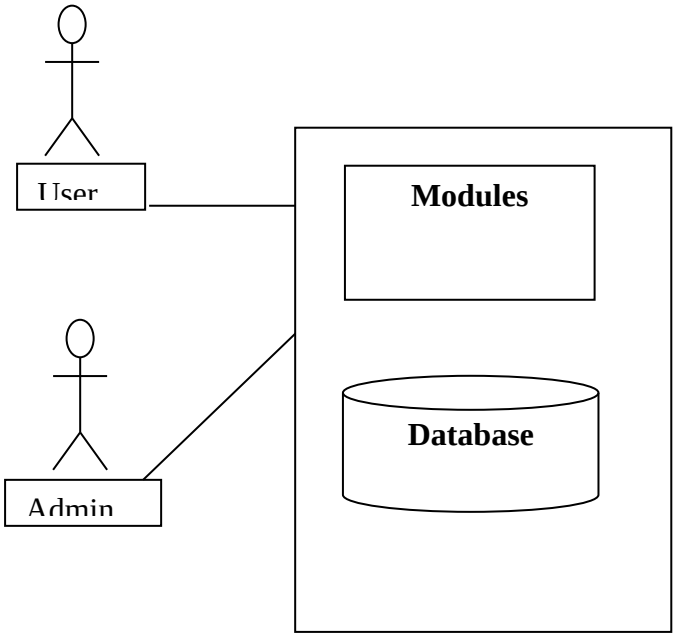
IEEE *Software Requirements Specifications.* IEEE Computer Society, 1998.

1.5. Overview of Document

The Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification.

2.0. Overall Description

2.1 System Environment

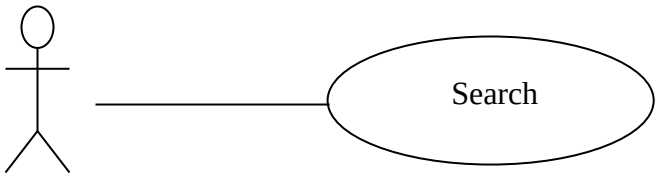


The **Cricket Training Management System** has two types of active actors one is general user and another is admin that cooperating system.

2.2 Functional Requirements Specification

This section outlines the use cases for each of the active operation. The user can search the information as per customer requirements and perform respective action.

Use case: Search



User

2.3 User Characteristics

The User is expected to be computer literate and be able to use a basic operation on system. The main screen of the system is displays different types of modules. User can select the required module and perform the tasks.

2.4 Non-Functional Requirements

The user's PC will contain a SQL database. SQL Server is already installed on this computer and is a Windows operating system.

3.0. Requirements Specification

3.1 External Interface Requirements

The only link to an external system is the link to the Database to verify the user of a System.

3.2 Functional Requirements

User can login the system. After verification they can access the main application of proposed system. Now user can open selected modules and input data correctly by checking validation criteria.

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data:

The logical structure of the data to be stored in the internal database is given in data structure

3.3.2 Security:

The PC on which the proposed system resides will have its own security. Only the admin will have physical access to the machine and the program on it. There is no special protection built into this system.

Bibliography

In course of development of the proposed system the information will be collected from the various documents and websites to synchronize the integrity of the contents present in the website. Not only this we have surveyed and collected the information of various cities, institutions and visiting sites and many more related destinations. I have also taken the help of so many books, magazines to collect the related information.

CHAPTER-7

RESULT AND CONCLUSION

This section discuses the result of the work done in this project and also mentions the future scope for improvement.

7.1 Conclusion

By designing this system, we are able to provide the basic functionality related to the submission activities with great ease. The use of C# technology has made it easier to design and develop the software architecture of this application. We were using the Microsoft Software Development Platform for the development of this project, which had given a complete, tight and integrated approach for the process of design and development of this project.

Hence, we may conclude that the application system being developed helps a great deal in modifying the computerized system.

7.2 Limitation of the system

Proposed Software system is based on IGNOU programme managment process. This is not suitable for other institution. This can run-on stand-alone environment only.

7.3 Future Scope of project

The project entitled **Proposed System** meets all requirements of criecket training programme management. It uses large database where large amount of data is stored.

An evaluation is carried out from the computer professional points of view. Design of the system and quality of the programming have to with stand the rigorous of careful assessment, programming standard is quite high today and a modular approach. Internal and external training in efficient program writing technique can achieve surprisingly good result. The organizations requirements change after certain interval of time and the system is required to be copyright. This software is suitable to solve the future problem. It does not require any special Hardware requirement, hence can be run on any common architect of computer.

.

Appendix A

List of Common Validations

1. No field in a form should be left blank.
2. The name of a person cannot be numeric value. It has to be in alphabets
3. Data has to be entered in a valid format.
4. Time has to be entered in valid format.
5. In any form, only one option can be selected from the given number of options

GLOSSARY Definition/Glossary

Data describe the numbers, text, graphics, image, and voice stored in this form that can be used by a computer.

Record – A set of related data items treated as a unit.

File – Set of record is termed as storage of files, and records in computer.

LAN Based Application – an application which run on internet or intranet.

Project Life Cycle (PLC) – Describe a way to make a successful project.

Configuration status reporting (CSR) – an activity that help software Developer to understanding what changes have been made and why.

Validation – Test to ensure that the software confirms to its requirement.

Prototyping – Creating a mock – up of an application.

References

1. Software Engineering, A Practitioner's Approach by **Roger S. Pressman**
2. An Introduction to Database Systems by **Bipin C Desai**
3. Professional C# 2nd edition by **Wrox.**
4. **Microsoft SQL server by NIIT**.