**TOURNAMENT REGISTRATION**

**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

Event management is the application of [project management](https://en.wikipedia.org/wiki/Project_management) to the creation and development of large scale events such as [festivals](https://en.wikipedia.org/wiki/Festival), conferences, ceremonies, formal parties, concerts, or [conventions](https://en.wikipedia.org/wiki/Convention_(meeting)). It involves studying the brand, identifying the [target audience](https://en.wikipedia.org/wiki/Target_audience), devising the event concept, and coordinating the technical aspects before actually launching the event.

The process of planning and coordinating the event is usually referred to as event planning and which can include budgeting, scheduling, site selection, acquiring necessary [permits](https://en.wikipedia.org/wiki/License), coordinating transportation and parking, arranging for speakers or entertainers, arranging decor, event security, [catering](https://en.wikipedia.org/wiki/Catering), coordinating with third party vendors, and emergency plans.

The events industry now includes events of all sizes from the [Olympics](https://en.wikipedia.org/wiki/Olympics) down to business breakfast meetings. Many industries, [charitable organizations](https://en.wikipedia.org/wiki/Charitable_organization), and interest groups hold events in order to market themselves, build business relationships, raise money, or celebrate achievement.

**1.2 INTRODUCTION**

Tournament registration System is the application of [project management](https://en.wikipedia.org/wiki/Project_management) to the creation and development of large scale events. Onlinesystem is used to register all the activity related to events. In any event, many organizers work simultaneously and it is very hard to manage these organizers. To manage all the activity we have developed this application. Any number of users can participate in any given event. In present system tournament organizers have to do all management work manually. They keep all information on papers. There is system which is meant for a particular organization. Keeping these entire problems in mind we have developed this system. This system helps the Tournament registration company to manage their paper work online and they can also retrieve report of last event they have completed.

**1.3 ABSTRACT**

This is an online tournament management system software project that serves the functionality of an event organizer. The system allows only registered users to login and new users are allowed to resister on the application. This is a web application developed in Asp.net and SQL. The project provides most of the basic functionality required for a tournament registration. It allows the user to view event details and post queries for organizers. Once the user is registered he can avail all the services provided by the application.

All this data is logged in the database and the organizer is responsible to manage all the events. And he has the privilege to reply for queries which are posted by the user as per requirement.

**CHAPTER 2**

**LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 TOURNAMENT**

Event Management System is useful for planning how to successfully organize a function. In order to organize an event team work is important and work allotted to each team member managed.

Event organizer is an application under project management for creating and developing festivals, events, etc. Through event management, users can study the workings of the product being looked at, identify the goal viewers, visualize the occasion idea, co-ordinate and plan the technical as well as logistical aspect of the event before executing the plan. An arrival on asset is essential for the occasion management industry, therefore careful planning and analysis is required for optimizing the return on investment. Since big events have a big impact on the country.

**2.2 REASON FOR CHOOSING .NET:**

**Limitations of C:**

* C developers are forced to contend with manual memory management.
* Ugly pointer arithmetic.
* C is structured programming language.
* Programmers require complete knowledge of best programming technique.

**Limitations of C++:**

* C++ can be thought as an Object Oriented layer on top of C.
* It involves manual memory management.
* Ugly pointer arithmetic.
* Ugly syntactical constructs.

**Limitations of JAVA/J2EE:**

* Java programmers must use java front to back during development cycle.
* It is not appropriate for many graphical or numerical intensive applications.

.NET provides solution to all the above mentioned problems.

**2.3 .NET FRAMEWORK**

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives: To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.

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

The .NET Framework has two main components:

* The common language runtime and
* The .NET Framework class library

The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

**2.4 INTRODUCTION TO ASP.NET**

ASP.NET is a unified Web development platform that provides the services necessary for you to build enterprise-class Web applications. While ASP.NET is largely syntax compatible with Active Server Pages (ASP), it provides a new programming model and infrastructure that allow you to create a powerful new class of applications. ASP.NET is part of the .NET Framework and allows you to take full advantage of the features of the common language runtime, such as type safety, inheritance, language interoperability, and versioning.

ASP.NET is supported on Windows 2000 (Professional, Server, and Advanced Server), Windows XP Professional, and the Windows Server 2003 family for both client and server applications.

In addition, to develop ASP.NET server applications, the following software is also required:

* Windows 2000 Server or Advanced Server with Service Pack 2, Windows XP Professional or 64-Bit Edition, or one of the Windows Server 2003 family products.
* MDAC 2.7 for Data.
* Internet Information Services.

**2.5 INTRODUCTION TO C#**

C# (pronounced as ‘C Sharp’) is a new computer-programming language developed by Microsoft Corporation, USA. C# is a fully object-oriented language like Java and is the first Component-oriented language. It has been designed to support the key features of .NET Framework, the new development platform of Microsoft for building component-based software solutions. It is a simple, efficient, productive and type-safe language derived from the popular C and C++ languages. Although it belongs to the family of C/C++, it is a purely object-oriented, modern language suitable for developing Web-based applications.

C# is designed for building robust, reliable and durable components to handle real-world applications. Major highlights of C# are:

* It is a brand new language derived from the C/C++ family.
* It simplifies and modernizes C++.
* It is the only component-oriented language available today.
* It is the only language designed for the .NET Framework.
* It is a concise, lean and modern language.
* It combines the best features of many commonly used languages: the productivity of Visual Basic, the power of C++ and the elegance of Java.
* It is intrinsically object-oriented and web-enabled.
* It has a lean and consistent syntax.
* It embodies today’s concern for simplicity, productivity and robustness.
* It will become the language of choice for .NET programming.
* Major parts of .NET Framework are actually coded in C#.

**2.6** [ADO.NET](http://ADO.NET)- DATABASE CONNECTIVITY

Most applications need [data access](http://www.startvbdotnet.com/aspsite/ado/default.aspx) at one point of time making it a crucial component when working with [applications.](http://www.startvbdotnet.com/aspsite/ado/default.aspx) Data access is making the application interact with a database, where all the data is stored. Different applications have different requirements for database access. [ASP.NET](http://ASP.NET) uses ADO .NET [(Active X Data Object)](http://www.startvbdotnet.com/aspsite/ado/default.aspx) as its data access and manipulation protocol which also enables us to work with data on the [Internet.](http://www.startvbdotnet.com/aspsite/ado/default.aspx)

[ADO.NET](http://ADO.NET) Data Architecture

Data Access in [ADO.NET](http://ADO.NET) relies on two components: Data Set and Data Provider.

1. Data Set

The dataset is a disconnected, in-memory representation of data. It can be considered as a local copy of the relevant portions of the database. The Data Set is persisted in memory and the data in it can be manipulated and updated independent of the database. When the use of this Data Set is finished, changes can be made back to the central database for updating. The data in Data Set can be loaded from any valid data source like [Microsoft SQL server](http://www.startvbdotnet.com/aspsite/ado/default.aspx) database, an Oracle database or from a Microsoft Access database.

2. Data Provider

The Data Provider is responsible for providing and maintaining the connection to the database. A Data Provider is a set of related components that work together to provide data in an efficient and performance driven manner. The .NET Framework currently comes with two Data Providers: the [SQL Data](http://www.startvbdotnet.com/aspsite/ado/default.aspx) Provider which is designed only to work with Microsoft's SQL Server 7.0 or later and the OleDb Data Provider which allows us to connect to other types of databases like Access and Oracle. Each Data Provider consists of the following component classes:

The Connection object which provides a connection to the database. The Command object which is used to execute a command. The Data Reader object which provides a forward-only, read only, connected record set. The Data Adapter object which populates a disconnected Data Set with data and performs update.

2.7 **INTRODUCTION TO SQL SERVER**

Microsoft SQL Server is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration.  In this article, we'll cover six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online.

Enterprise Manager is the main administrative console for SQL Server installations.  It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network.  You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick and dirty method for performing queries against any of your SQL Server databases.  It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database.  You can monitor many different event types and observe database performance in real time.  SQL Profiler allows you to capture and replay system "traces" that log various activities.  It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.

Service Manager is used to control the MSSQLServer (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQLServerAgent processes.  An icon for this service normally resides in the system tray of machines running SQL Server.  You can use Service Manager to start, stop or pause any one of these services.

Data Transformation Services (DTS) provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats.  The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

**2.8 INTERNET INFORMATION SERVER [IIS]**

IIS server includes a broad range of administrative features for managing web sites and the web server. With programmatic features like ASP, ASP.NET, you can create and deploy scalable, flexible web applications.

**Features of IIS:**

Support of Internet standards:

The features include HTTP support and standard internet services such as:

* **World Wide Web (WWW):** It supports HTTP allowing users to publish content to the internet. Files can be placed in folders in our websites so the users can view with a web browser. We can install business applications and publish in HTML.
* **File Transfer Protocol:** It is a industry standard protocol used for transferring files between computers on a TCP/IP network. FTP enables to use one computer to host multiple domain names.
* **Simple Mail Transfer Protocol (SMTP):** It is a standards-based, secure and scalable email server. It supports a distributed email server using SMTP and Post Office Protocol (POP3).

**CHAPTER 3**

**SYSTEM ANALYSIS**

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

The existing system is an intranet based application which is used by a network of only particular organization. It takes more effort and physical space to keep track of paper documents, to find information and to keep details secure. When mistakes are made or changes or corrections are needed, often a manual transaction must be completely redone rather than just updated. With manual or partially automated systems information often has to be written down and copied or entered more than once.

**3.2 LIMITATIONS OF EXISTING SYSTEM**

It is clear from the details of existing system that the whole system is complex and required a lot of manual work.

* Less Efficient
* Time consuming
* Less Reliable
* Lack of customized services
* Expensive

**3.3 PROBLEM STATEMENT**

There is no application which handles all the organizations together to maintain events online.

**3.4 PROPOSED SYSTEM**

In any event, many organizers work simultaneously and it is very hard to manage these organizers. To manage all the activity we have developed this application. The Tournament registration system is useful the user to provide information regarding the event that are conducted irrespective of organizations. The **event manager** here admin is the person who plans and executes the event, taking responsibility for the creative, technical, and logistical elements. In the proposed system, the administrator of the application manages all the event types and views all the registered organizer and users. Admin also has privilege to view all the feedbacks given users. The organizer after registration and updates his profile. Organizers will add events to the event list along with its details; also if organizer is interested in other events taking place then he can also participate in that. The registered user can register for individual events and participate in all the events.

**3.5 ADVANTAGES**

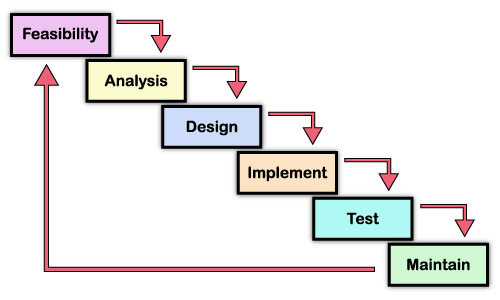
* The system is useful as it calculates an exact cost for all the resources required during the event.
* The user gets all the resources at a single place instead of wandering around for these.
* This system is effective and saves time and cost of the users.

**3.6 APPROACH ADAPTED:**

According to Software Engineering the approach adopted to develop this project is the Iterative waterfall Model. The iterative waterfall Model is a systematic approach that begins at the feasibility study phase and progress through analysis, design, coding, testing, integration and maintenance. Feedback paths are there in each phase to its preceding phase as show in the fig to allow the correction of the errors committed during a phase that are detected in later phase.

**Feasibility study:**

The main aim of this phase is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all the relevant information relating to the product such as different data items which would be input to the system, processing required to be carried out on these data, the output data required to be produced by the system, as well as constraints on the behavior of the system.



**Fig: Waterfall Model**

**Requirement Analysis and Specification:**

The main aim of this phase is to understand the exact requirements of the customer and to document them properly.

**Design:**

The goal of design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. In technical terms, during the design phase the software architecture is derived from SRS document. Two distinctly different design approaches are available: the traditional approach and the object oriented approach. We have adopted traditional design to develop the product.

**Coding:**

Once design is complete, goal of the coding phase is to translate the design of the system into code in a given programming language. For a given design, the aim in this phase is to implement the design in the best possible manner. We have coded the design using C# language to develop the product.

**Testing:**

Testing is the major quality control measure employed during software development. Its basic function is to detect errors in the software.

**Maintenance:**

Maintenance is not a part of software development. It is an extremely important activity in the life of software product. Maintenance involves performing any one or more of the following kinds of activities:

* Correcting errors that were not discovered during the product phase. This is called corrective maintenance.
* Improving the implementation of the system, and enhancing the functionalities of the system according to the customer’s requirements. This is called perfective maintenance.
* Porting the software in a new environment. This is called adaptive maintenance.

**CHAPTER 4**

**REQUIREMENT ANALYSIS**

**CHAPTER 4**

**REQUIREMENT ANALYSIS**

**SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

Software Requirement Specification is the starting point of the software development activity. It includes an introduction that gives the purpose, scope and an overview of the system. This needs requirement by talking to the people and understanding their needs. It also includes a general description of the product perspective, product function and certain user characteristics of the system. It also specifies the overall functional requirements, performance requirements and design constraints. The SRS is a means of translating the idea in the mind of the clients (the input), into a formal document (the output of the requirement phase). The Software Requirement Specification document is organized in such a manner it aids validation and system design.

**4.1 OVERALL DESCRIPTION**

**4.1.1 USER INTERFACE**

There are 3 actors in this system.

They are:

1. Admin

2. Organizer

3. User

**4.1.2 HARDWARE INTERFACES**

Processor : Pentium 4 +

RAM : 2GB

Hard Disk : 20GB

Speed : 1.2 GHz+

**4.1.3 SOFTWARE INTERFACES**

Operating System : Windows XP or Higher

IDE : Visual Studio 2010

Language : C#

Framework : ASP.NET 4.0

Back End : MS SQL Server

**Other Web Technologies:** HTML, CSS, JAVASCRIPT, AJAX, JQUERY.

**4.1.4 COMMUNICATIONS INTERFACES**

The HTTP protocol will be used to facilitate communications between the client and server.

**4.2 SPECIFIC REQUIREMENTS:**

**Functional Requirements:**

1. **User class – Administrator**

* **Login Module**

Admin will login to the application by providing id and password. He is the owner of the application and he manages the entire application.

* **Manage Event Type**

Event types are managed by the administrator such that all the event types can be added, updated, or deleted.

* **View Event Details**

Details of all the events can be viewed in this module.

* **View Registered Organizer**

Organizers who are responsible to perform many of the activities in the application their details can be viewed in this module.

* **View Registered User**

Users who avails services of the application their details can be viewed in this module. User will be able to access the application after getting registered.

* **View Feedback**

Feedbacks can be viewed in this module which are posted by the user

1. **User class – Organizer**

* **Registration**

Organizers can get registered to the application by providing suitable credentials

* **Login Module**

Organizers can successfully login to the application by providing id and password which was given during registration process.

* **Update Profile**

Profiles of organizer can be updated in this module

* **Manage Events**

Events can be added along with its details in this module.

* **View the Events of Others**

Events can be viewed in this module if interested he can participate in the events

* **View Queries and Reply**

Queries can be viewed and replied.

1. **User class – User**

* **Registration**

User will have to register first for the application after registration login can be performed

* **Login Module**

Login functionality involves entering unique id and password after logging in one can access and avail all the services

* **Update Profile**

Profiles of user can be updated in this module

* **Get Register for Events**

For each event user have to register themselves.

* **View Event Details**

Details of each event can be viewed here in this module.

* **Post Queries for Organizer**

Queries of organizers can be posted here in this module.

* **Post Feedback**

Feedbacks can be posted here by user to the organizer

**4.3 COMMUNICATION INTERFACE REQUIREMENTS**

The Hyper Text Transfer Protocol (HTTP) will be used to facilitate communications between the client and server.

* HTTP protocol is used to transmit documents around network.
* Mainly the website is developed using ASP.NET as front and SQL Server 2005 as backend.
* The website is based on three tier architecture with data server, application server and a client.

**CHAPTER 5**

**SYSTEM DESIGN**

**CHAPTER 5**

**SYSTEM DESIGN**

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases particularly testing and maintenance.

The design activity often results in three separate outputs –

* Architecture design.
* High level design.
* Detailed design.

**Architecture Design:**

Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. The focus is on identifying components or subsystems and how they connect. In other words, the focus is on what major components are needed.

**High Level Design:**

In high level design identifies the modules that should be built for developing the system and the specifications of these modules. At the end of system design all major data structures, file format, output formats, etc., are also fixed. The focus is on identifying the modules. In other words, the attention is on what modules are needed.

**Detailed Design:**

In the detailed design the internal logic of each of the modules is specified. The focus is on designing the logic for each of the modules. In other words how modules can be implemented in software is the issue.

A design methodology is a systematic approach to creating a design by application of a set of techniques and guidelines. Most methodologies focus on high level design.

* 1. **ARCHITECTURAL DESIGN**

In this project three tier architecture is used.

**Introduction:**

As a [developer](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##), the .NET framework and Visual Studio present many choices for choosing the right architecture, from placing the data access code directly in the UI through datasets and data source controls, to creating a data access layer that talks to the [database](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##), all the way to creating an n-tier architecture approach that consists of multiple layers, and use data-transfer objects to pass data back and forth.

**Layer:**

A layer is a reusable portion of code that performs a specific function. In the .NET environment, a layer is usually setup as a project that represents this specific function. This specific layer is in charge of working with other layers to perform some specific goal. In an application where the [presentation](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##) layer needs to extract information from a backend database, the presentation would utilize a series of layers to retrieve the data, rather than having the database calls embedded directly within itself. Now we will look briefly at the latter situation first.

**5.1.1. Two-Tier Architecture**

When the .NET 2.0 framework became available to the world, there were some neat features that allowed the developer to connect the framework’s GUI controls directly to the database. This approach is very handy when rapidly developing applications. However, it’s not always favorable to embed all of the business logic and data access code directly in the [web site](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##), for several reasons:

* Putting all of the code in the web site (business logic and data access) can make the application harder to maintain and understand.
* Reusing database queries in the presentation layer often isn’t done, because of the typical data source control setup in the [ASP](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##).NET framework.
* Relying on the data source controls can make [debugging](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##) more difficult, often due to vague error messages.

So in looking for an alternative, we can separate the data access code and business logic into separate “layers”.

* + 1. **Three tier Architecture:**

Three tier architecture consists of three layers. They are:

**The Data Layer:**

The key component to most applications is the data. The data has to be served to the presentation layer somehow. The data layer is a separate component (often setup as a separate single or group of projects in a .NET solution), whose sole purpose is to serve up the data from the database and return it to the caller. Through this approach, data can be logically reused, meaning that a portion of an application reusing the same query can make a call to one data layer method, instead of embedding the query multiple times. This is generally more maintainable.

**Business Layer:**

Though a web site could talk to the data access layer directly, it usually goes through another layer called the business layer. The business layer is vital in that it validates the input conditions before calling a method from the data layer. This ensures the data input is correct before proceeding, and can often ensure that the outputs are correct as well. This validation of input is called business rules, meaning the rules that the business layer uses to make “judgments” about the data.

One of the best reasons for reusing logic is that applications that start off small usually grow in functionality. The business layer helps move logic to a central layer for “maximum reusability.”

**Presentation Layer:**

The ASP.NET web site or windows forms application (the UI for the project) is called the presentation layer. The presentation layer is the most important layer simply because it’s the one that everyone sees and uses. Even with a well structured business and data layer, if the presentation layer is designed poorly, this gives the users a poor view of the [system](http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx##).

**Three tier Architecture:**

The **presentation tier** contains the UI (User Interface) elements of the site, and includes all the logic that managers the interaction between the visitor and the client’s business. (ASP.NET Web Forms, Web User Controls, ASP.NET Master Pages)

The **business tier** receives requests from the presentation tier and returns a result to the presentation tier depending on the business logic it contains. (C# Classes)

The **data tier** is responsible for storing the application’s data and sending it to the business tier when requested. (SQL Server Stored Procedures).



**Fig- Three tier architecture**

**5.2 HIGH LEVEL DESIGN**

**5.2.1 Data Flow Diagram**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD).

**Symbols used in DFD’s:**

**Processes:**

A process transforms data values. The lowest processes are our functions without side effects.

**Data Flows:**

A data flow connects the output of an object or process to the input of another object or process. It represents the intermediate data values within the computation. It is draws as an arrow between the procedure and the consumer of the data value. The arrow is labeled with the description of the data, usually its name or type.

**Actors:**

An actor is an active object that drives the data flow graph by producing or consuming values. Actors are attached to the inputs and the outputs of a dataflow graph. In sense, the actors lie on the boundary of the flow graph but terminate the flow of data as sources and sinks of data, and so are sometimes called terminators.

**Data Store:**

A data store is a passive object within a data flow diagram that stores data for later access. Unlike an actor, a data store does not generate any operations on its own but merely responds to requests to store and access data.

**Context data flow diagram:**

It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the 'Level 0 DFD') the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.

ORGANIZER

View and manage

ADMIN

Manage Serves

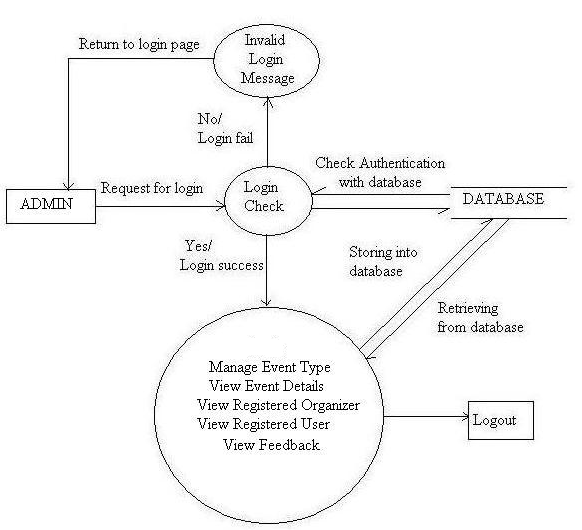
USER

**Fig: Context flow diagram**

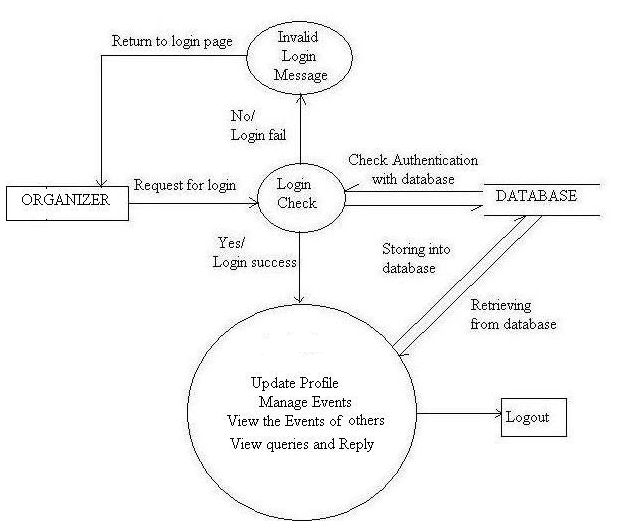
This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

**Level 1 (high level diagram)**

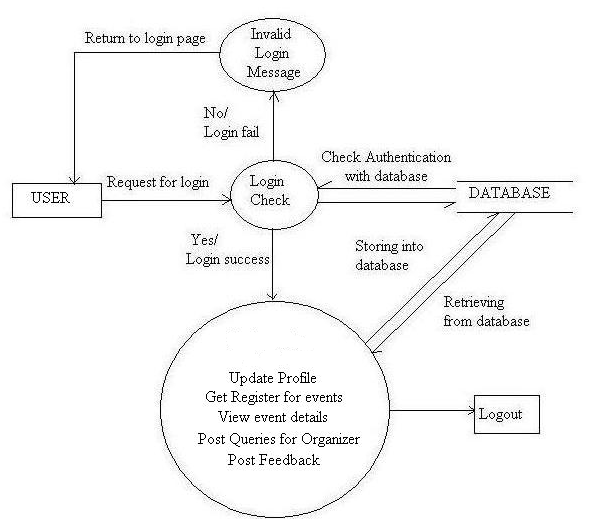
This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major and high-level processes of the system and their interrelation. A process model will have one, and only one, level-1 diagram. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level1.



**Fig: DFD for Admin**

****

**Fig: DFD for Organizer**



**Fig: DFD for User**

* 1. **DETAILED DESIGN**

**Use Case Diagram**

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

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

Interaction among actors is not shown on the use case diagram. If this interaction is essential to a coherent description of the desired behavior, perhaps the system or use case boundaries should be re-examined. Alternatively, interaction among actors can be part of the assumptions used in the use case.

**Use cases:**

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

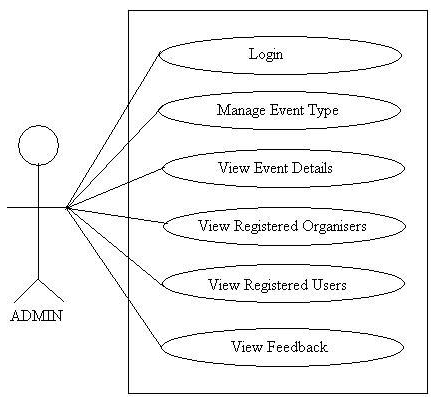
**Actors:**

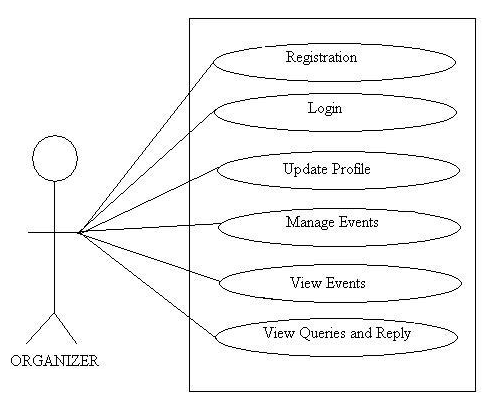
An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

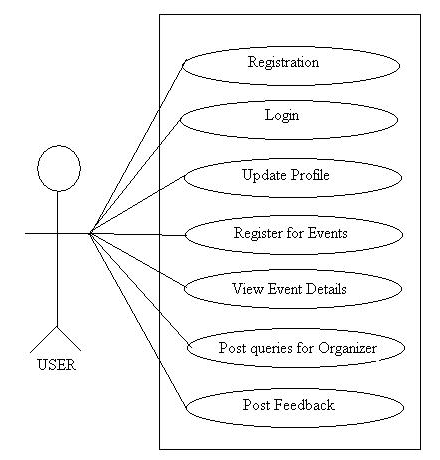
**System boundary boxes:**

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not.

**Use Case Diagram**







**5.4 SEQUENCE DIAGRAM**

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It describe interactions among classes in terms of an exchange of messages over time. Sequence diagrams are used to show how objects interact in a given situation. An important characteristic of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom

**Targets/Class roles/State:**

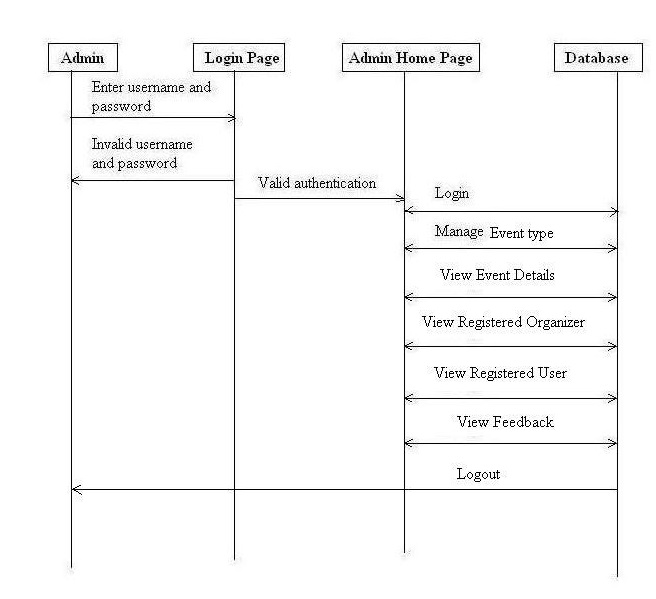
Objects as well as classes can be targets on a sequence diagram, which means that messages can be sent to them. A target is displayed as a rectangle with some text in it. Below the target, its lifeline extends for as long as the target exists. Targets can be actor, boundary, control, entity and database.

**Messages:**

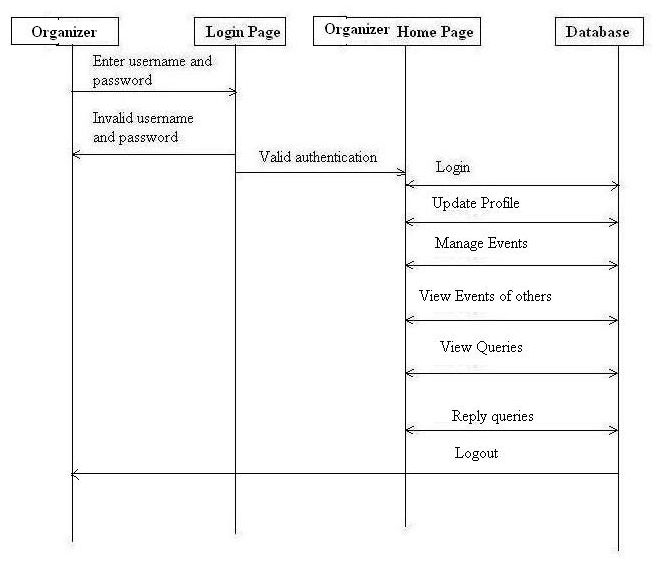
Messages are arrows that represent communication between objects.

**Lifelines:**

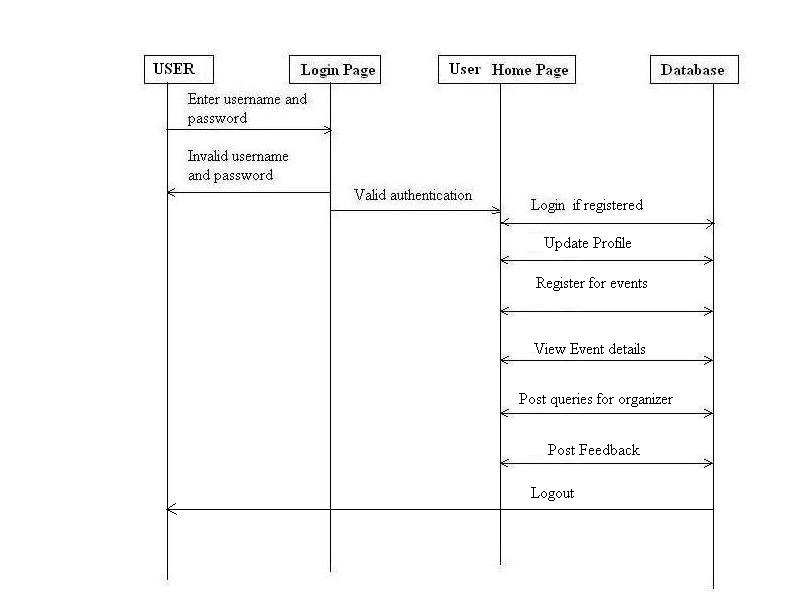
Lifelines are vertical dashed lines that indicate the object's presence over time.



**Fig: Sequence diagram for Admin**



**Fig: Sequence diagram for Organizer**



**Fig: Sequence diagram for User**

**5.5 ER DIAGRAM**

An entity is a business object that represents a group, or category of data.

ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of a information system. Diagrams created using this ER-modeling technique are called Entity-Relationship Diagrams, or ER diagrams or ERDs. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases.

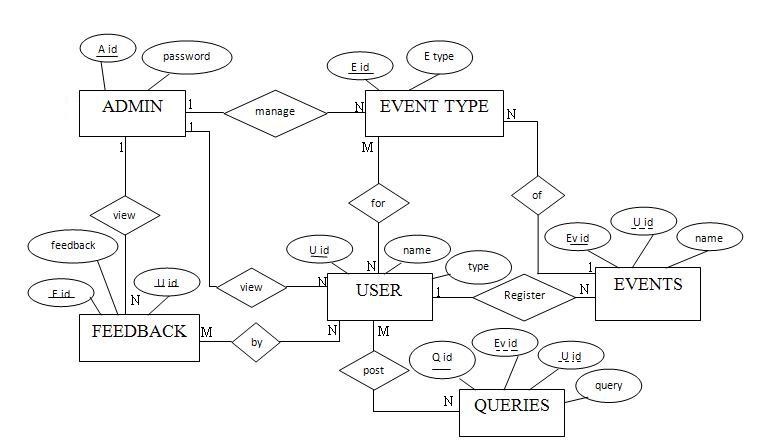
Dr. Peter Chen is the originator of the Entity-Relationship Model. His original paper about ER-modeling is one of the most cited papers in the computer software field. Currently the ER model serves as the foundation of many system analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems.

The original notation for ER-Diagrams uses rectangles to represent entities, and diamonds to represent relationships.

There are three basic elements in ER-Diagrams:

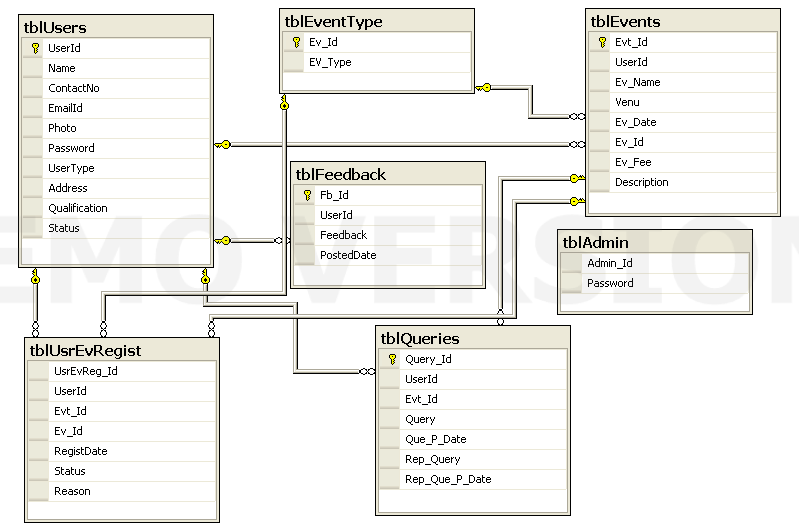
* Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.
* Attributes are the data we want to collect for an entity.
* Relationships describe the relations between the entities.

ERDs show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.



**Fig: ER diagram**

**5.5 RELATIONS ESTABLISHED IN DATABASE TABLES**



**Fig: Database diagram**

**CHAPTER 6 IMPLEMENTATION**

**CHAPTER 6**

**IMPLEMENTATION**

**Features of Object Oriented Paradigm:**

This web application is implemented using object oriented programming language. Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

**Features of Object Oriented paradigm:**

* Emphasis is on data rather than procedure.
* Programs are divided into what are known as objects.
* Data structures are designed such that they characterize the objects.
* Methods that operate on the data of an object are tied together in the data structure.
* Objects may communicate with each other through methods.
* New data and methods can be easily added whenever necessary.
* Follows bottom-up approach in program design.
* Data is hidden and cannot be accessed by external functions.

This project is implemented using three tier architecture. ASP.NET is used in the presentation layer, C# classes are used in the Business logic, Table adopter is used in the data tier and MS SQL server 2005 (database) is used as the backend.

**Implementation Steps:**

**Presentation Layer** is Asp.net (front end) which invokes the Business logic through button click or page load event or SelectedIndexChange event of the dropdownlist.

**Business Logic** contains the common methods. An object for Business logic class is created and object will invoke the method.

The business logic object will call table Adopter method. Table Adopter will open the database connection. Since SQL server 2005 is used as the backend, to interact with the database SqlDataSource is used.

**ADD Coding Here…….**

**CHAPTER 7**

**TESTING**

**CHAPTER 7**

**TESTING**

**7.1 TESTING PURPOSE**

Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

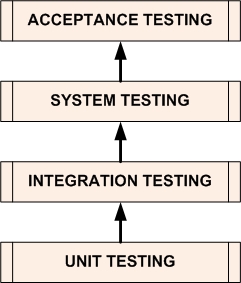
* Finding defects which may get created by the programmer while developing the software.
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product.

Software testing is performed to verify that the completed software package functions according to the expectations defined by the requirements/specifications. The overall objective to not to find every software bug that exists, but to uncover situations that could negatively impact the customer, usability and/or maintainability.

From the module level to the application level, this article defines the different types of testing. Depending upon the purpose for testing and the software requirements/specs, a combination of testing methodologies is applied. One of the most overlooked areas of testing is regression testing and fault tolerant testing.

**7.2 LEVELS OF TESTING:**

There are four **levels of software testing.**

software_testing_levels1

1. **Unit Testing:**

**It is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.**

1. **Integration Testing:**

It is a level of the software testing process 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.

1. **System Testing:**

It is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.

1. **Acceptance Testing:**

It is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery.

**7.3 REGRESSION TESTING:**  
 Regression testing is retesting sub-systems/modules/units to insure that modifications to one sub-system/module/unit does not cause unexpected results in another sub-system/module/unit. This is also known as ripple effect testing.

**Why is Regression Testing Necessary?**  
 Regression testing is necessary because many times modifications in one part of the code cause unexpected problems in a "totally unrelated" area of the code.

**7.4 TYPES OF TESTING**

**White box testing:**

**It is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions. Also known as structural testing and Glass box Testing.**

**Black box testing:**

Internal system design is not considered in this type of testing. Tests are based on requirements and functionality. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

**CHAPTER 8**

**CONCLUSION AND FUTURE ENHANCEMENT**

**CHAPTER 8**

**CONCLUSION AND FUTURE ENHANCEMENT**

**8.1 CONCLUSION**

Many tournament organizers are looking for innovative ways to streamline their workload and at the same time grow their event's prestige and overall functional satisfaction.  While the growing trend is to take all aspects of event management online with event registration software, many tournament organizers are still wary that that their participants will be scared away at the thought of submitting their data manually. So there is a need of an online tournament registration application.  Online entry is a fast and seamless experience for a participant linking to the form from your official event website.  The online registration form can have the same look and feel as your site so that the registrant feels like he has never left the website.  An online registrant can quickly fill out a simple online form that has been custom tailored to only collect the information required of a person entering their category and submit it to your database within minutes. To manage all the activity we have developed this application. Any number of users can participate in any given event. This system helps the event management company to manage their paper work online and they can also retrieve report of last event they have completed.

**8.2 FUTURE ENHANCEMENT**

This application in future can be enhanced in many ways. Some are listed below.

* While registering, we can send SMS notifications to their mobile numbers registered.
* We can enhance by confirming the generated OTP- One Time Password for registration purpose.
* We can include module to ask a user to change password frequently.
* When the users log any queries, we can send the status to their mobile through SMS.
* We can even send news as alerts to the users mobile.

**CHAPTER 9**

**ANNEXURE**

**CHAPTER 9**

**ANNEXURE**

**SCREEN SHOTS**

**CHAPTER 10**

**REFERENCES**

**CHAPTER 10**

**REFERENCES**

**REFERRED WEBSITES**

**www.wikipedia.org**

**www.msdn.microsoft.com/asp.net**

**www.csharphelp.com**

**www.w3schools.com**

**www.dotnetspider.com**

**www.csharpcorner.com**

**www.codeguru.com**

**www.techopedia.com**

**www.msdn.microsoft.com/net/quickstart/aspplus/default.com**

**www.asp.net**

**www.fmexpense.com/quickstart/aspplus/default.com**

**www.asptoday.com**

**www.aspfree.com**

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

**www.smartdraw.com/resources/tutorials/context-diagrams**

[**www.edrawsoft.com/Data-Flow-Diagram-Symbols.php**](http://www.edrawsoft.com/Data-Flow-Diagram-Symbols.php)

**www.tutorialspoint.com/uml/uml\_use\_case\_diagram.htm**

[**www.uml-diagrams.org/**](http://www.uml-diagrams.org/)

[**www.lucidchart.com/pages/ER-diagram-symbols-and-meaning**](https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning)

**www.righthandtech.com/software-testing.php**

**REFERRED BOOKS**

**“Programming C#, Fourth Edition- Building .NET Applications with C#” -By Jesse Liberty**

**“Working With Microsoft Visual Studio 2005 Team System” -By Richard Hundhausen**

**“Begin ASP.NET 2.0 with visual C#.NET”, Wrox, By Chris Ullman.**

**Software Engineering, Ian Summerville, Sixth Edition, Pearson Education Ltd, 2001**

**Database Management System by Raghu Ramakrishnan**