**A**

**MCA MAJOR PROJECT REPORT**

**ON**

“**ONLINE CRIME DETECTION”**

IN

THE PARTIAL FULFILLMENT OF REQUIREMENT FOR THE AWARD OF THE DEGREE

OF

MASTER OF COMPUTER APPLICATIONS

**Under the guidance of: Submitted by:**

Mr. Naveen Nagpal Shivangi Monga

Asst. Professor RollNo-4395371

Department Of Computer MCA 6th Semester

Science & Applications



Department of Computer Science & Application

P.D.M. College of Engineering, Bahadurgarh

(Affiliated to Maharishi Dayanand University)

Rohtak- 124001

**CANDIDATE’S DECLARATION**

I Shivangi Monga hereby declare that the work presented in the project report entitled “ONLINE CRIME DETECTION” submitted to Department of Computer Science & Applications, P.D.M College Of Engineering., affiliated to Maharishi Dayanand University, Rohtak for the partial fulfillment of the award of degree of “**Master of Computer Applications**” is an authentic record of my work carried out during the final semester, 2015 at XTREME DIGITAL SOLUTIONS under the supervision of

Mr.Sanjay Rajan Padhy (External Guide Information) and Internal Guide as

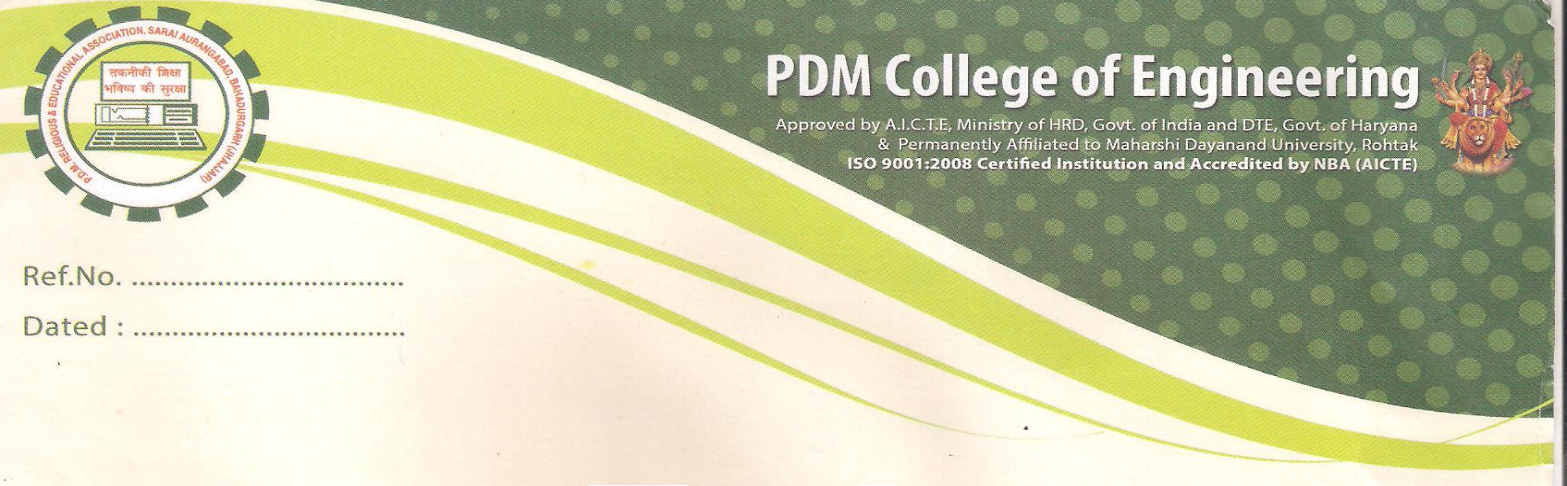
Mr. Naveen nagpal Department of Computer Science & Applications , PDM College Of Engineering,Bahadurgarh, Jhajjar.

The matter embodied in this project report has not been submitted elsewhere by anybody for the award of any other degree.

Shivangi Monga

Rollno-4395371

MCA 6th semester

****

##### CERTIFICATE

This is to certify that the project titled "**Online Quiz**" is a bonafide work carried out by **Ms. Shilpa Verma, Roll No. 1259096** in the partial fulfillment of the requirement for the award of the degree of **Master of Computer Applications** from P.D.M. College of Engineering, Bahadurgarh, Haryana under the supervision of **Mr. Jitender Kathuria.**

Mr. Jitender Kathuria Prof.(Dr.)S. Srinivasan

Asst. Professor Head of Department

Deptt. Of Computer Sci. & App. Deptt. Of Computer Sci &App.

P.D.M.College of Engineering P.D.M.College of Engineering

****

**ACKNOWLEDGEMENT**

I would like to thank people who were part of this work in numerous ways. In particular, I wish to thank Mr.Sanjay Rajan Padhy (External Guide Information), my project guides for their suggestions and improvements in this project and providing continuous guidance at each and every stage of the project.

Thanks are extended especially to my guide Mr.Naveen Nagpal (Designation, P.D.M. College of Engineering) and to Prof.(Dr.)S. Srinivasan (HOD, Department of Computer Science & Applications, P.D.M. College of Engineering).I also thank Director Sir of P.D.M. College of Engineering for his valuable support. I must also be thankful to my classmates and friends for their continuous co-operations and help in completing this project.

Last but not the least, I want to express my thanks to my parents and family members for their support at every step of life.

Shivangi Monga

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Contents** | **Page No.** |
| **1** | Introduction |  |
| **1.1** | **Company Profile** |  |
| **1.2** | **Abstract of the Project** |  |
| **1.3** | **Title of the Project** |  |
| **1.4** | **Objective** |  |
| **1.5** | **Problem Specifications/Need of the Project** |  |
|  |  |  |
| **2** | Feasibility Study |  |
| **2.1** | **Existing System** |  |
| **2.2** | **Proposed System** |  |
| **2.3** | **Brief Feasibility Study** |  |
| **2.4** | **Requirement Specification** |  |
|  |  |  |
| **3** | Software Requirement Specification System Analysis |  |
| **3.1** | **Introduction** |  |
| **3.2** | **Selection of Technology/Specific Requirements** |  |
| **3.3** | **System Flowchart** |  |
|  |  |  |
| **4** | System Design |  |
| **4.1** | **E.R Diagram** |  |
| **4.2** | **Data Flow Diagram(0 & 1 Level Diagram)** |  |
| **4.3** | **Modules** |  |
| **4.4** | **Database** |  |
| **4.5** | **Input & Output Form(Screen Layout)** |  |
|  |  |  |
| **5** | Coding (If Any) |  |
|  |  |  |
| **6** | **Implementation/Technological Environment** |  |
| **7.** | **Testing & Results** |  |
|  |  |  |
| **8.** | **System Securities and Maintenance** |  |
|  |  |  |
| **9.** | **Conclusion & Future Enhancement** |  |
|  |  |  |
| **10.** | **Bibliography** |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **CHAPTER1**  **INTRODUCTION**   * 1. **COMPANY PROFILE**   **Microsoft Corporation** is an American  [technology company](http://en.wikipedia.org/wiki/Technology_company) headquartered in [Redmond](http://en.wikipedia.org/wiki/Redmond,_Washington), [Washington](http://en.wikipedia.org/wiki/Washington_(state)), that develops, manufactures, licenses, supports and sells [computer software](http://en.wikipedia.org/wiki/Computer_software), [consumer electronics](http://en.wikipedia.org/wiki/Consumer_electronics) and [personal computers](http://en.wikipedia.org/wiki/Personal_computers) and services.  download.jpg  Its best known software products are the [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) line of [operating systems](http://en.wikipedia.org/wiki/Operating_systems), [Microsoft Office](http://en.wikipedia.org/wiki/Microsoft_Office) [office suite](http://en.wikipedia.org/wiki/Office_suite), and [Internet Explorer](http://en.wikipedia.org/wiki/Internet_Explorer) [web browser](http://en.wikipedia.org/wiki/Web_browser). Its flagship hardware products are the [Xbox](http://en.wikipedia.org/wiki/Xbox) game consoles and the [Microsoft Surface](http://en.wikipedia.org/wiki/Microsoft_Surface) tablet lineup. It is the [world's largest software maker](http://en.wikipedia.org/wiki/List_of_the_largest_software_companies) measured by revenues. It is also one of the [world's most valuable companies](http://en.wikipedia.org/wiki/List_of_corporations_by_market_capitalization).  Microsoft was founded by [Bill Gates](http://en.wikipedia.org/wiki/Bill_Gates) and [Paul Allen](http://en.wikipedia.org/wiki/Paul_Allen) on April 4, 1975, to develop and sell [BASIC](http://en.wikipedia.org/wiki/BASIC) [interpreters](http://en.wikipedia.org/wiki/Interpreter_(computer_software)) for [Altair 8800](http://en.wikipedia.org/wiki/Altair_8800). It rose to dominate the [personal computer](http://en.wikipedia.org/wiki/Personal_computer) [operating system](http://en.wikipedia.org/wiki/Operating_system) market with [MS-DOS](http://en.wikipedia.org/wiki/MS-DOS) in the mid-1980s, followed by [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows). The company's 1986 [initial public offering](http://en.wikipedia.org/wiki/Initial_public_offering), and subsequent rise in its share price, created three billionaires and an estimated 12,000 millionaires from Microsoft employees. Since the 1990s, it has increasingly diversified from the operating system market and has [made a number of corporate acquisitions](http://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions_by_Microsoft). In May 2011, Microsoft acquired [Skype Technologies](http://en.wikipedia.org/wiki/Skype_Technologies) for $8.5 billion in its largest acquisition to date.  As of 2013, Microsoft is market dominant in both the [IBM PC-compatible](http://en.wikipedia.org/wiki/IBM_PC-compatible) operating system and [office software suite](http://en.wikipedia.org/wiki/Office_software_suite) markets (the latter with [Microsoft Office](http://en.wikipedia.org/wiki/Microsoft_Office)). The company also produces a wide range of other software for desktops and servers, and is active in areas including [Internet search](http://en.wikipedia.org/wiki/Web_search_engine) (with [Bing](http://en.wikipedia.org/wiki/Bing)), the [video game industry](http://en.wikipedia.org/wiki/Video_game_industry) (with the [Xbox](http://en.wikipedia.org/wiki/Xbox_(console)), [Xbox 360](http://en.wikipedia.org/wiki/Xbox_360) and [Xbox One](http://en.wikipedia.org/wiki/Xbox_One) consoles), the digital services market (through [MSN](http://en.wikipedia.org/wiki/MSN)), and mobile phones (via the [Windows Phone](http://en.wikipedia.org/wiki/Windows_Phone) OS). In June 2012, Microsoft entered the personal computer production market for the first time, with the launch of the [Microsoft Surface](http://en.wikipedia.org/wiki/Microsoft_Surface), a line of [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer).  With the acquisition of Nokia's devices and services division to form [Microsoft Mobile Oy](http://en.wikipedia.org/wiki/Microsoft_Mobile), the company re-entered the Smartphone hardware market, after its previous attempt, [Microsoft Kin](http://en.wikipedia.org/wiki/Microsoft_Kin), which resulted from their acquisition of [Danger](http://en.wikipedia.org/wiki/Danger_(company)). While jointly developing a new OS with IBM in 1984, [OS/2](http://en.wikipedia.org/wiki/OS/2), Microsoft released [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), a graphical extension for MS-DOS, on November 20, 1985 moved its headquarters to Redmond on February 26, 1986, and on March 13 the company went [public](http://en.wikipedia.org/wiki/Public_company) the ensuing rise in the stock would make an estimated four billionaires and 12,000 millionaires from Microsoft employees. Due to the partnership with IBM, in 1990 the [Federal Trade Commission](http://en.wikipedia.org/wiki/Federal_Trade_Commission) set its eye on Microsoft for possible collusion; it marked the beginning of over a decade of legal clashes with the U.S. Government. Microsoft released its version of OS/2 to [original equipment manufacturers](http://en.wikipedia.org/wiki/Original_equipment_manufacturer)(OEMs) on April 2, 1987 meanwhile, the company was at work on a [32-bit](http://en.wikipedia.org/wiki/32-bit) OS, [Microsoft Windows NT](http://en.wikipedia.org/wiki/Microsoft_Windows_NT), using ideas from OS/2; it shipped on July 21, 1993, with a new modular [kernel](http://en.wikipedia.org/wiki/Kernel_(computing)) and the [Win32](http://en.wikipedia.org/wiki/Windows_API) [application programming interface](http://en.wikipedia.org/wiki/Application_programming_interface) (API), making [porting](http://en.wikipedia.org/wiki/Porting) from [16-bit](http://en.wikipedia.org/wiki/16-bit) (MS-DOS-based) Windows easier. Once Microsoft informed IBM of NT, the OS/2 partnership deteriorated.  **CONTENT DEVELOPMENT AND TECHNICAL WRITING**:  We have written multiple technical books for our US Clients. Some of the books which we have written and are being used in training for US companies are:   * OOP Programming using Java * J2EE Programming * Introduction to Struts * .NET framework * ASP.NET * XML * Web services  |  | | --- | | **1.2 ABSTRACT OF THE PROJECT** |   Title of Project : Online Crime Detection  Status : Subscribe and unsubscribe the services  Team Size : One Team Leader  Server Name : SQL server  **Background and Related Work:**   * Creating Web pages. * Making all the respective modules.   **Hardware and Software Requirements:**   1. Processor : Core i3 Ram : 8GB Hard-disk : 1 TB 2. OS : Windows 8/8.1 Front tool : Visual studio 2013 Back End : SQL Server 2008 Technology : ASP.net 4.0 Language : VISUAL C#  Platform : .NET Framework 4.0   **Client’s Requirements** :  According to the objectives the clients requirements are as under :  Providing an online comprehensive solution to manage quiz  where the individuals are participating in a team.   |  |  | | --- | --- | |  | **1.3 TITLE OF THE PROJECT**  The project titled as “Online Crime Detection “ is a web based application. This software provides facility for reporting online crimes, complaints, missing persons, show most wanted person details mailing as well as chatting. Any Number of clients can connect to the server. Each user first makes their login to sever to show their availability.  This will help institutes as:   * There will be no need to get new software every time to conduct an online test.   Also like other online websites, it will help students by:   * Saving the extra time of going to far away Exam Centre. * Students need not wait for their results.   Also this website will remove the flaws of existing Manual Systems like:   * Reducing the manual labour (Decreases Overheads). * Avoiding Mistakes Due To Human Error (Accurate). * Will Increase Efficiency and Save Time. * Will Allow Neat Handling Of Data Rather Than Error Prone Records.   The institutes will register themselves with a unique login name and password; the unique id will be issued to the institutes by the website.  After login:   * They will enter exam details like number of questions, +ve and -ve marks. * Then they will enter the questions along with the answers which can later be deleted and edited. * Also they will enter the list of eligible candidates with their id names which can also be edited later. * Institutes will be able to view the students list along with their respective results.   Also for students:   * They should be able to login with their id, name and institute id. * They should be able to give the exam as per the details entered by respective institutes.   Also they should be able to view their score after test  Finishes.  **WHAT IS M3hQuiz.COM?**  M3hQuiz.com simply contains the information of six subjects of computer science i.e. computer basis, computer networking, java, php, database, c#, operating system. By using it we can test our knowledge in the given subjects without any hesitation.it is a secure medium to get test to check our mind. Here each user has a unique user name and password. Only after entering password he can give test.no other user cannot get information about any other. There are 20 questions in the test in which user have to attempt 10 correct answers to get passed, otherwise he will fail. User can check his test record anytime even after many years.  **1.4 OBJECTIVE**   * Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. * It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. * When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow   **TECHNOLOGIES USED**   * **.NET:**  It consists of two parts * CLR stands for Common language runtime, which is run time environment to execute the .Net programs. * NET API (Application Programming Interface) that consists of inbuilt classes used in .Net programs. * **VC#.Net** * .NET is Platform Dependent, Secure, Object Oriented, component oriented, Scalable, and * **Robust Programming Language** * ADO.NET (Active X data objects) is an API, which is used for the communication of .NET programs with different databases. * ASP.NET (Active Server Pages) is also a part of Dot Net and is also used for server-side programming but with the help of ASP.Net we can segregate the work of a web designer and a developer. * SQL Server is used as database, used to store data. It is RDBMS. * **Scripting Languages** * HTML, XML ,DHTML-CSS, JavaScript, DOM   **1.5 PROBLEM SPECIFICATION / NEED OF**  **THE PROJECT** | |  | * **not all students can use the computer very well** * **Technology isn't always reliable. Information**   **can be lost if a system breaks down.**  Need for this kind of service came in existence because   * **The application will help in saving the time for the user.** * **It gives you the option of online FIR and details of criminals.** * **The system gives total security to the user .It restricts**   **the unauthorized access .**   * **This application provides better facility to the candidates**   **who login to the system.**  **CHAPTER 2**  **FEASIBILITY STUDY** | |  |  | |  |

**FEASIBILITY STUDY**

A feasibility study is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation.

A feasibility study is a test of a system proposal according to its workability impact on organization, ability to meet user needs and effective use of resources. The objective of a feasibility study is not to solve a problem but to acquire a sense of its scope. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project. During the study, the problem definition is crystallized and the aspects of the problem to be included in the system are determined. After the initial investigation of system, this helped to have in-depth study of existing system, understanding its strength and weaknesses & requirements for new proposed system.

|  |  |
| --- | --- |
|  | **2.1 EXISTING SYSTEM**  In the existing system only we can see the details of particular information about the police stations in our state, the existing system has more workload for the authorized person, but in the case of Proposed System, the user can registered in our site and send the crime report and complaint about a particular city or person.  **ADVANTAGES OF EXISTING SYSTEM**   * More man power. * Time consuming. * Consumes large volume of pare work. * Needs manual calculations. * No direct role for the higher officials. * Damage of machines due to lack of attention.   **2.2 PROPOSED SYSTEM**    The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.  **2.3 BRIEF FEASIBILITY STUDIES & METHODOLOGY**  An important outcome of the preliminary investigation is the determination that the system requested is feasible. There are three aspects in the feasibility study.  C:\Documents and Settings\Administrator\My Documents\My Pictures\v4.bmp   * **TECHNICAL FEASIBILTY:**   This is concerned with specifying equipment and software that will successfully satisfy the user’s requirements. The technical needs of the system may vary considerably, but might include:   1. The facility to produce output is in a given time. 2. Response time under certain conditions. 3. Ability to process a certain volume of transaction at a particular speed. 4. Facility to a communicate date to distance location.   At the feasibility stage it is desirable that two or three different configuration will be pursued that satisfy the key technical requirements but which represent different levels of ambitions and cost. Investigation of these technical alternatives can be aided by approaching a range of suppliers for preliminary discussion. Out of all types of feasibility, technical feasibility generally is the most difficult to determine.   * **OPERATIONAL FEASIBILITY:**   Operational feasibility determines how much effort will go in the proposed information system, and in educating and training the employee on the new system, along with the new ways of conducting the business. Operational study strives on ensuring that the equilibrium of the organization and status quo in the organization are not disturbed and changes are readily accepted by the users.  Following question helped to us test the operational feasibility of system.   1. What changes will be brought with the system? 2. What organizational structures are disturbed? 3. What new skill will be required? Do existing staff members have these skills? If not, can they be trained in due course of time?  * **ECONOMICAL FEASIBILTY:**   Economical analysis is the most frequently used technique for evaluation the effectiveness of a proposed system. More commonly known as cost/benefit analysis: the procedure is to determine the benefits and savings that expected from a proposed system and compare with them costs. If benefits outweigh cost, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to make if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system lifecycle. This feasibility also depends upon quality of a staff hired and the proposed duration of time taken in this project sometimes it might be possible due to extension phase to phase of the project development.   * **METHODOLOGY:**   The following approach is used to design this system, which is called “classic lifecycle” or “waterfall model”. This linear sequential model suggests a systematic, sequential approach to software development that begins at the system’s level and progress through analysis, design, coding, testing and maintenance.   1. System/Information engineering and modeling. 2. Software requirement analysis. 3. Design. 4. Code generation. 5. Testing. 6. Maintenance.   C:\Documents and Settings\Administrator\My Documents\My Pictures\v5.bmp |

**2.4 REQUIREMENT SPECIFICATIONS**

**Hardware used while developing application**

Processor : Core 3.

Ram : 8 GB

Hard-disk : 1 TB

**Software used while developing application**

OS : windows 8/8.1

Front tool : Visual Studio 2103

Back End : SQL Server 2008

Technology : ASP.NET

Language : VISUAL C#

Platform : .NET Framework 4.0

**Minimum Hardware requirement at SERVER SIDE**

Processor : 1GHZ

RAM : 256 MB

HARD-DISK : 10 GB space (for software and application)

**Minimum Software Requirement at SERVER SIDE**

OS : Win XP sp2  
Web-Server : Internet information server

Software : visual web developer 2008

Platform : .NET Framework 3.5

**Minimum HARDWARE and SOFTWARE Required at CLIENT level**

Any browser which is XHTML 1.0 compatible like Microsoft internet 6.0, Firefox, Opera, Google Chrome

Any hardware which support above mentioned browser.

**Requirement specification** plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The software requirements specification document enlists all necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with the project team and customer.

 A requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints) .

The SRS is a means of translating the ideas in the minds of the clients (the input), into formal document (the output of the requirements phase). Thus, the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input has none of these properties.

**Functional Requirements:**

* The system should have a provision for the Customer to use the System 24x7x365 days
* System should generate a unique id.
* System should have facilitate the customer to Register and create new account.
* The system should have an option for the customer to track the status
* System should have a facility for the Administrator to view all the complaints by date.
* The system should facilitate the users to register complaints on all the work done on the issue.
* The system should provide an option for the Consultants to send the response regarding solution to the problem/status.
* The system should have a provision for the Administrator to add/update/delete the users.
* The system should have a provision for the Administrator to add categories and subcategories.

**Non-Functional Requirements:**

* The users of the system should be provided user id and password along with the well defined access privileges.
* 24X7 internet connectivity should be provided for well functioning of the system.
* Systems should be provided with proper backup media and resources to handle system crash scenarios.
* Systems must have to be user friendly and easy to use.
* The System should provide necessary provisions to protect information and system resources with respect to confidentiality and integrity.
* System should be easily maintainable.

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION SYSTEM ANALYSIS**

|  |
| --- |
|  |
|  |
|  |

**System Analysis**

This system has an advantage over the manual system that existed prior to the designing of the system. Earlier this system service subscribe and unsubscribe was taking place via at any place.

The existing system has some limitation

1. Less manual work
2. Error occurs frequently.
3. Data redundancy & inconsistency.

The existing system has a database and so they do not have a possibility of losing the data about the parties involved in transaction that is nominee & transfer.

**Requirement Analysis**

Software Requirement Specification (SRS) is a condition of capability needed by a user to solve problem or achieve an objective. The requirements phase translate the ideas in the minds of the clients (the output), in to a formal document (the output of the requirement phase) thus the output phase is a set of formally specified requirement, which hopefully are complete and consistence, while the output has none of these properties. Generally the SRS is a document that completely describes what the proposed software should do without describing how the software will do it. The basic goal of the requirements phase is to produce the SRS, which describe the complete external behavior of the proposed software. However producing the SRS is easier said than done. A basic limitation for this is that the user need keeps changing as the requirement in which the system was to functioning changes with time. This happens more in complex application where all the needs may not be known to any set of people during the requirement phase. The objective SRS phase is to specify what is needed from the system, not how the system will provide it.

* 1. **DATA FLOW DIAGRAMS**

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

* **DFD SYMBOLS:**

In the DFD, there are four symbols.

1. A square defines a source (originator) or destination of system data.
2. An arrow identifies data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

* **CONSTRUCTING A DFD:**

Several rules of thumb are used in drawing DFD’s:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized

**0 LEVEL D.F.D**:

This is the context level D.F.D. 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.

**1 LEVEL D.F.D.**

This D.F.D. shows all the processes together with all the data stores (tables). It shows the true data flow i.e. how data is actually flowing in the system. Data is coming from which table and going into which table is clearly shown by this DFD. This DFD is the main reference for the development of the system. After understanding the whole system, the application developer will fall back upon this DFD during the Development phase

**2Level (D.F.D)**

There can be one level 2 DFD for each process of the Level 1 DFD.  Level 2 shows a process broken down into greater detail.  Level 2 Diagrams are only necessary where the Level 1 process is more complex, and where the particular process is relevant to the analysis.

**3.2 SELECTION OF TECHNOLOGY/SPECIFIC REQUIREMENTS ENTITY RELATIONSHIP DIAGRAMS**

**THE .NET FRAMEWORK**

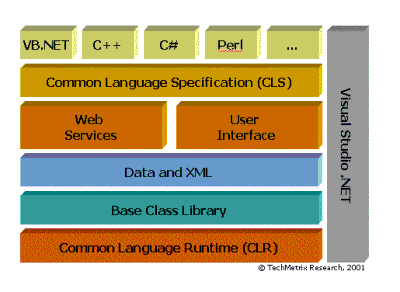
A frame work is commonly thought of as a set of class libraries that aid in the development of applications. The .net framework is more than just a set of classes. The .net framework is targeted by compliers using a wide variety of applications. Including everything from small components that run on handheld devices to large Microsoft ASP.ET application that span web farms, where multiple web serves act together to improve the performance fault tolerance of a web site. The .NET framework is responsible for providing a basic platform that these applications can share. This basic platform includes a runtimes set of services that oversee the execution of applications. A key responsibility of the runtime is to manage execution so that software written by different programming languages uses classes and other types safely.

**MICROSOFT.NET FRAMEWORK ARCHITECTURE**

Microsoft's .NET Framework is comprised of two main components - the Common Language Runtime (CLR) and the .NET Framework class libraries. The CLR is the real foundation of the .NET Framework. It is the execution engine for all .NET applications. Every target computer requires the CLR to successfully run a .NET application that uses the .NET Framework. The main features of CLR include:

* Automatic Memory Management
* Thread Management
* Code Compilation & Execution
* Code Verification
* High level of security
* Remoting
* Structured Exception Handling
* Interoperability between Managed and Unmanaged code.
* Integration with Microsoft Office System

All .NET applications are compiled into Intermediate Language code (MSIL). When executed on the CLR, MSIL is converted into native machine code specific to the operating platform. This process is done by a Just in Time (JIT) compiler. The code executed by the CLR is called as Managed Code. This code is type safe and thoroughly checked by the CLR before being deployed. The .NET runtime also provides a facility to incorporate existing COM components and DLL's into a .NET application. Code that is not controlled by the CLR is called Unmanaged Code.

The .NET Framework is further comprised of Common Type System (CTS) and Common Language Specification (CLS). The CTS defines the common data types used by .NET programming languages. The CTS tells you how to represent characters and numbers in a program. The CLS represents the guidelines defined by for the .NET Framework. These specifications are normally used by the compiler developers and are available for all languages, which target the .NET Framework. 

.Net architecture

**COMMON LANGUAGE SPECIFICATION**

To fully interact with other objects regardless of the language they were implemented in, objects must expose to callers only those features that are common to all the languages they must interoperate with. For this reason, the Common Language Specification (CLS), which is a set of basic language features needed by many applications, has been defined. The CLS rules define a subset of the Common Type System; that is, all the rules that apply to the common type system apply to the CLS, except where stricter rules are defined in the CLS. The CLS helps enhance and ensure language interoperability by defining a set of features that developer can rely on to be available in a wide variety of languages. The CLS also establishes requirements for CLS compliance; these help you determine whether your managed code conforms to the CLS and to what extent a given tool supports the development of managed code that uses CLS features.

If your component uses only CLS features in the API that it exposes to other code (including derived classes), the component is guaranteed to be accessible from any programming language that supports the CLS. Components that adhere to the CLS rules and use only the features included in the CLS are said to be CLS-compliant components.

The CLS was designed to be large enough to include the language constructs that are commonly needed by developers, yet small enough that most languages are able to support it. In addition, any language constructs that makes it impossible to rapidly verify the type safety of code was excluded from the CLS so that all CLS-compliant languages can produce verifiable code if they choose to do so.

**COMMON LANGUAGE RUNTIME**

The Common Language Runtime (CLR) is the virtual machine component of Microsoft's .NET initiative. It is Microsoft's implementation of the Common Language Infrastructure (CLI) standard, which defines an execution environment for program code. The CLR runs a form of byte code called the Microsoft Intermediate Language (MSIL), Microsoft's implementation of the Common Intermediate Language.

Developers using the CLR write code in a high level language such as C# or VB.Net. At compile-time, a .NET compiler converts such code into MSIL (Microsoft Intermediate Language) code. At runtime, the CLR's just-in-time compiler (JIT compiler) converts the MSIL code into code native to the operating system. Alternatively, the MSIL code can be compiled to native code in a separate step prior to runtime. This speeds up all later runs of the software as the MSIL-to-native compilation is no longer necessary.

Although some other implementations of the Common Language Infrastructure run on non-Windows operating systems, the CLR runs on Microsoft Windows operating systems.

The virtual machine aspect of the CLR allows programmers to ignore many details of the specific CPU that will execute the program. The CLR also provides other important services, including the following:

* Memory management
* Thread management
* Exception handling
* Garbage collection
* Security

**INTRODUCTION TO ASP.NET**

Although so Microsoft Visual Basic.NET is a powerful but simple language aimed primarily at developers creating web applications for the Microsoft .NET platform. It inherits many of the best features of C++ and Microsoft Visual Basic, but with some of the inconsistencies and anachronisms removed, resulting in cleaner and logical language. VB also contains a variety of useful new innovations that accelerate application development, especially when used in conjunction with Microsoft Visual Studio .NET.

The Common Language Runtime provides the services that are needed for executing any application that’s developed with one of the .NET languages. This is possible because all of the .NET languages compile to a common Intermediate Language. The CLR also provides the common type system that defines those data types that are used by all the .Net languages. That way, you can use same data types regardless of what.NET language you’re using to develop your application presentations.

**ASP.NET:**  Microsoft, realizing that ASP does possess some significant shortcomings, developed ASP.net. ASP.net is a set of components that provide developers with a framework with which to implement complex functionality. Two of the major improvements of ASP.net over traditional ASP are scalability and availability. ASP.net is scalable in that it provides state services that can be utilized to manage session variables across multiple web services in a server farm. Additionally, ASP.net possesses a high performance process model that can detect application failures and recover from them. We use the fundamentals of programming with VB using Visual Studio .NET and .NET framework.

The project is the starting point for authoring applications, components &services in Visual Studio.NET 2005.It eats as a container that manages your source code, data connections & references. A project is organized as part of a solution, which can contain multiple projects that are independent of each other. C# project file has .asproj extension whereas solution file has .sln extension.

In order to write code against an external component, your project must first contain a reference to it. A reference can be made to the following types of component.

(1) .NET class libraries or assemblies

(2) COM components

(3) Other class libraries of projects in the same solution

(4) XML web services

**FEATURES OF ASP.NET:**

(1) Component Infrastructure.

(2) Language Integration.

(3) Internet Interoperation.

(4) Simple Development.

(5) Simple Deployment.

(6) Reliability.

(7) Security

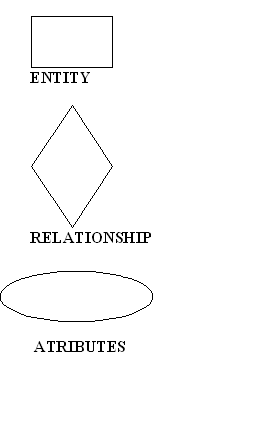
* **E-R DIAGRAMS:**

An **entity**-**relationship** (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database.

It shows the logical flow of the system. For a system, it describes the input (source) , output (destination) ,database (data stores) and procedures (data flows) all in a format that meets the user’s requirement. When analyst prepares the logical system design, he specifies the user’s needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. The logical design also specifies input forms and screen layouts.

The activities following logical design are the procedures followed in the physical design e.g. producing programs, software, file, and a working system. Design Specifications instruct the user about what the system should do.

To construct ER-diagrams, we use:

****

Data models are tools used in analysis to describe the data requirements and assumptions in the system from a top-down perspective. They also set the stage for the design of databases later on in the SDLC.

There are three basic elements in ER models:

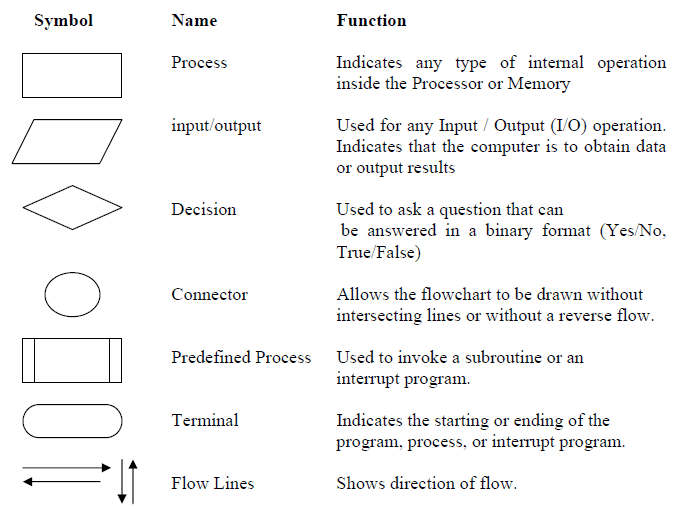
1. Entities are the "things" about which we seek information.

2. Attributes are the data we collect about the entities.

3. Relationships provide the structure needed to draw information from multiple entities.

**3.3 SYSTEM FLOWCHART**

A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.Flowcharts are used in designing and documenting complex processes or programs.

****

|  |  |  |
| --- | --- | --- |
|  |  |  |

start

end

Display result

Compute result

Question=20&time=10min

Display question

Randomize question

Generate question

eligible

authenticate

stop

Display message

**CHAPTER 4**

**SYSTEM DESIGN**

After the analysis phase we have with us the details of the existing system and the requirements of the user for the new system. This phase diverts focus from the problem domain to the solution domain.

It acts as a bridge between the requirement phase and its solution. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. Emphasis is on translating performance specifications into design specifications.

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.

There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

Object-oriented analysis and design (OOAD) methods are becoming the most widely used methods for computer system design. The UML has become the standard language used in Object-oriented analysis and design.

It is widely used for modelling software systems and is increasingly used for high designing non-software systems and organizations.

**External Design**

* External design consists of conceiving, planning out and specifying the externally observable characteristics of the software product.
* These characteristics include user displays or user interface forms and the report formats, external data sources and the functional characteristics, performance requirements etc.
* External design begins during the analysis phase and continues into the design phase.

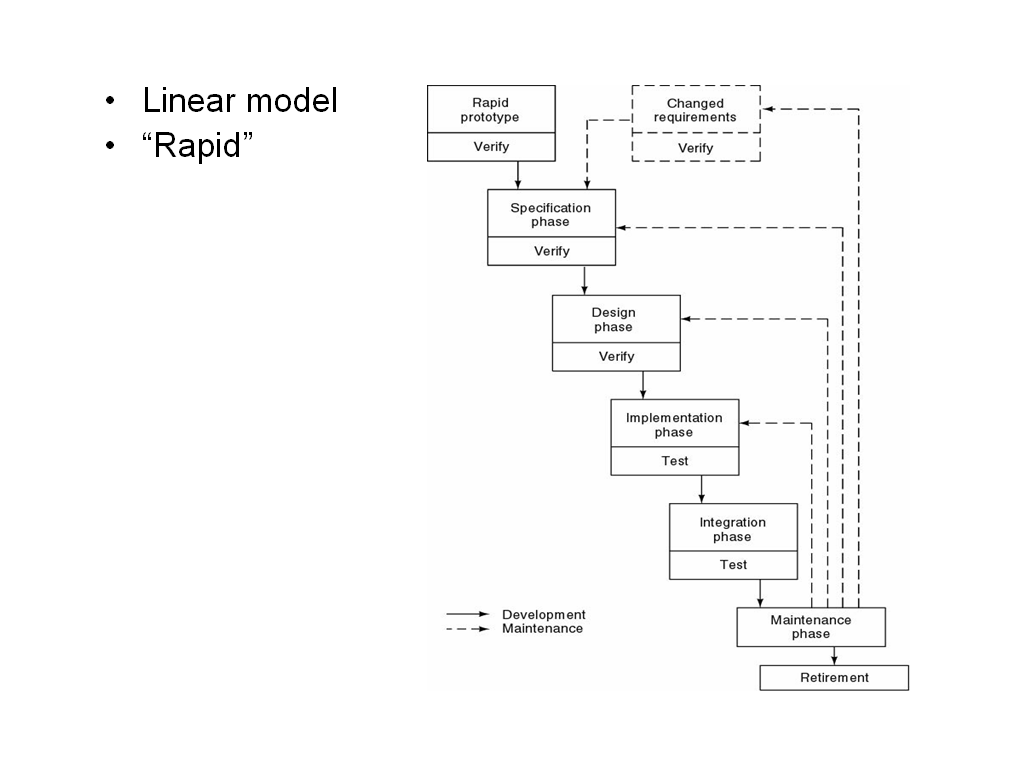
**Physical design**

* The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified/ authenticated, how it is processed, and how it is displayed as output.
* Physical design, in this context, does not refer to the tangible physical design of an information system.
* To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc.
* It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc.

**DESIGN NOTATIONS**

**Design Methodology: Rapid Application Development (RAD)**

* Rapid Application Development (RAD) is a methodology in which a systems designer produces prototypes for an end-user.
* The end-user reviews the prototype, and offers feedback on its suitability. This process is repeated until the end-user is satisfied with the final system.



* 1. **Data Flow Diagram**

**0 LEVEL D.F.D**.

This is the context level D.F.D. 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 proposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system.

User ID, Complaints Complaint Manager

Administrator

User

Solutions, Feedback

**1ST LEVEL D.F.D.**

This D.F.D. shows all the processes together with all the data stores (tables). It shows the true data flow i.e. how data is actually flowing in the system. Data is coming from which table and going into which table is clearly shown by this DFD. This DFD is the main reference for the development of the system. After understanding the whole system, the application developer will fall back upon this DFD during the Development phase.

**Level 1 –Administration**

Registration details

Administrator

User name, Password View user details

View complaints

Complaints

Login

Online crime detection

Add solutions

Feedback

View Feedback

User

Add FIR

FIR

**2nd Level (D.F.D)**

There can be one level 2 DFD for each process of the Level 1 DFD.  Level 2 shows a process broken down into greater detail.  Level 2 Diagrams are only necessary where the Level 1 process is more complex, and where the particular process is relevant to the analysis.

**Level 2 –Administration**

FIR No: Details

FIR Details

Complaint Details

Administrator

User

Reports

FIR

Complaints

**Level 1- User:-**

Verify

Complaints

Username/Password

Username/Password

User

Registration

Login

Complaints

**Level 2- User:-**

Solutions

Complaint Details

Complaint Status

Complaint ID

User name Password

User

Complaint Status

Feedback

Complaints

**4.2 E.R Diagram**

Receive

Register

USER

Registered case

Registration

**USE CASE DIAGRAM:-**

The Use Case Diagram is a represent of a user’s interface and behavior of a system or a part of a system and description of a set of sequences of actions, including variants that a system perform to the observable result of value to an actor.

A use case diagram are describes a set of sequences in which sequence represent by the interaction of the outside to the system with the system itself.

These behaviors are effect system level function that uses to visualize, specify, construct and document the intended behaviors of the system.

The use case diagram are used to the user requirements of a system and It can also internal and external factors influencing of the system.

Login

View FIR

Register FIR

Registration

View criminal details

**User Admin**

Feedback

**GANTT CHART:-**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Week1** | **Week2** | **Week3** | **Week4** | **Week5** | **Week6** | **Week7** | **Week8** |
| **Requirement Gathering** |  |  |  |  |  |  |  |  |
| **Design** |  |  |  |  |  |  |  |  |
| **Test Cases** |  |  |  |  |  |  |  |  |
| **Coding** |  |  |  |  |  |  |  |  |
| **Quality Assurance** |  |  |  |  |  |  |  |  |
| **Testing** |  |  |  |  |  |  |  |  |
| **Build** |  |  |  |  |  |  |  |  |

**PERT CHART:-**

**1**

**2**

**4**

**6**

**3**

**5**

**8**

**7**

Create Schedule

Programming

Test Code

Test System

User Test

Buy Hardware

Installation

* Numbered rectangles are nodes and represent event or milestones.
* Directional arrow represent dependent task that must be completed sequentially.
* Diverging arrows directions indicate possibly concurrent tasks.

# 4.3 MODULES

It contains two main modules:-

* **Admin: - Admin** is responsible to change all information about crime. After login he can see registered users, their FIR records and the database. Admin can insert new user also can edit &delete existing users.

**Admin has following parts or module**

* View and reply user complaint: This module helps the admin to view and reply user’s complaint details
* View and reply user crimes: This module helps the admin to view and reply user’s crimes details
* New admin: This module used for add new admin
* Add and delete latest news: This module helps the admin to add and delete latest news.
* View and delete user’s feedback: This module helps the admin to add and delete user’s feedback
* Add, delete and view most wanted persons: This module helps the admin to add, delete and view most wanted person details
* Add, delete and view missing persons: This module helps the admin to add, delete and view missing person details
* Add and view Criminal registration: This module helps the admin to add and view criminal registrations
* Add and view FIR: This module helps the admin to add and view Fir reports

**User: - U**ser is responsible for giving test. If he is a registered user only then he can give test. Otherwise he has to registration before giving test. After registration he is given a unique user name and secure password. After login he can complaint any of the case of his choice among the given.

**User module has following parts or pages:**

* View news: This module help the visitor to see the latest news which can updated by the admin.
* The registered user module includes,
* Add Complaint: This module help the user to report online complaints.
* Add Crime report: This module help the user to report online crimes.
* Edit Complaint: This module help the user to edit his complaint details.
* Edit Account: This module help the user to update his or her profile.
* View complaint status: This module allows us to view the status of all complaint that you have posted earlier.
* View crime status: This module allows us to view the status of the all crimes that you have posted earlier.

**4.4Database Design**

A database is a collection of inter-related data stored with a minimum of redundancy to serve many applications. It minimizes the artificially embedded in using separate files.

The primary objectives are fast response time to enquire more information at low cost, control of redundancy, clarity and ease to uses, accuracy and fast recovery. The organization of data in a database aims to achieve three major objective, they are data integration, data integrity and data independence. During the design of the database almost care has been taken to keep up the objectives of the database design.

**DATA STRUCTURE**

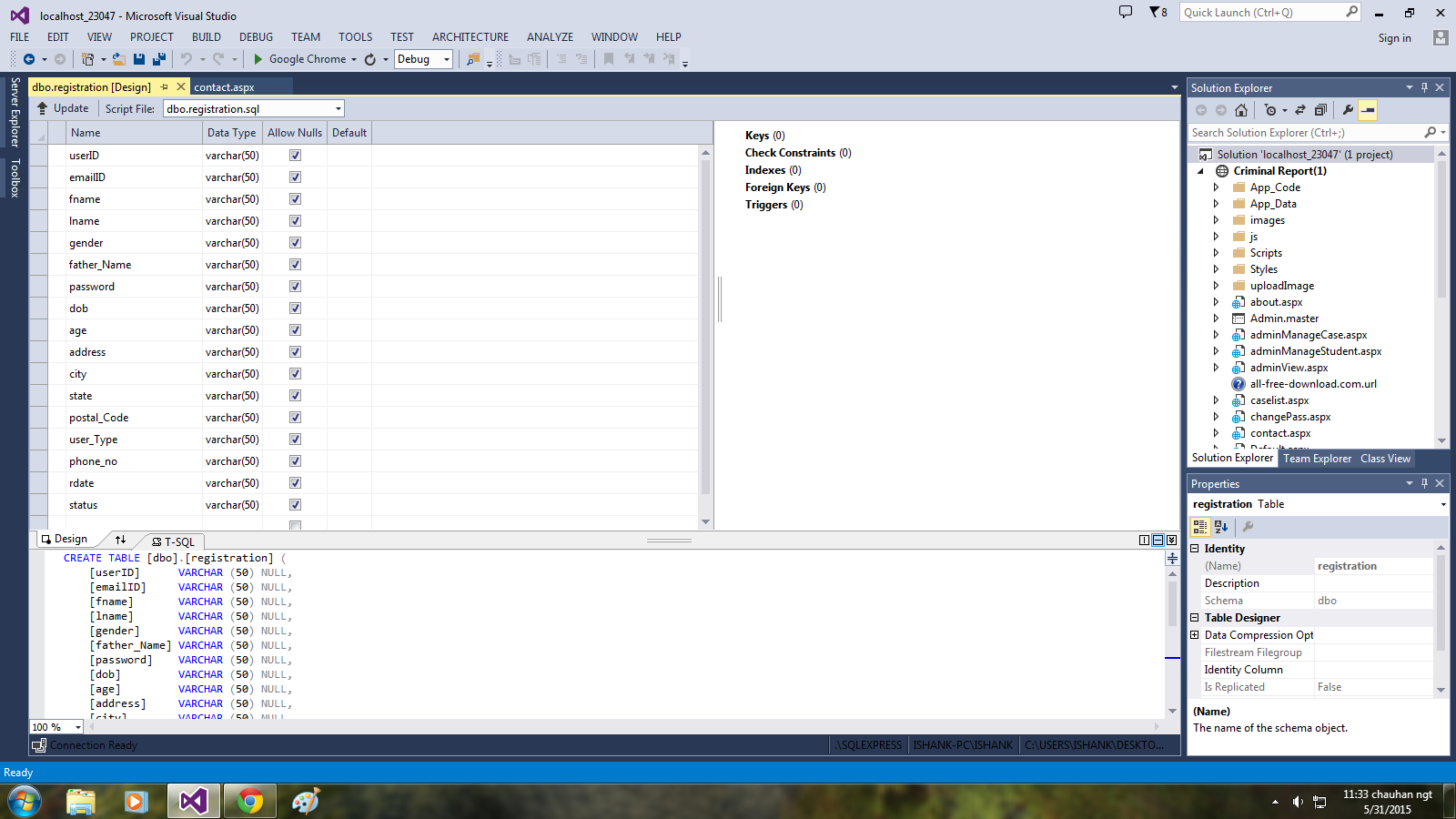
This part of the design consists of the overall database schema or we can say that table which consists various types of records. Table of a database consists attributes, entities, and topples for storing and manipulating records.

The tables that are used in this project are as follows:

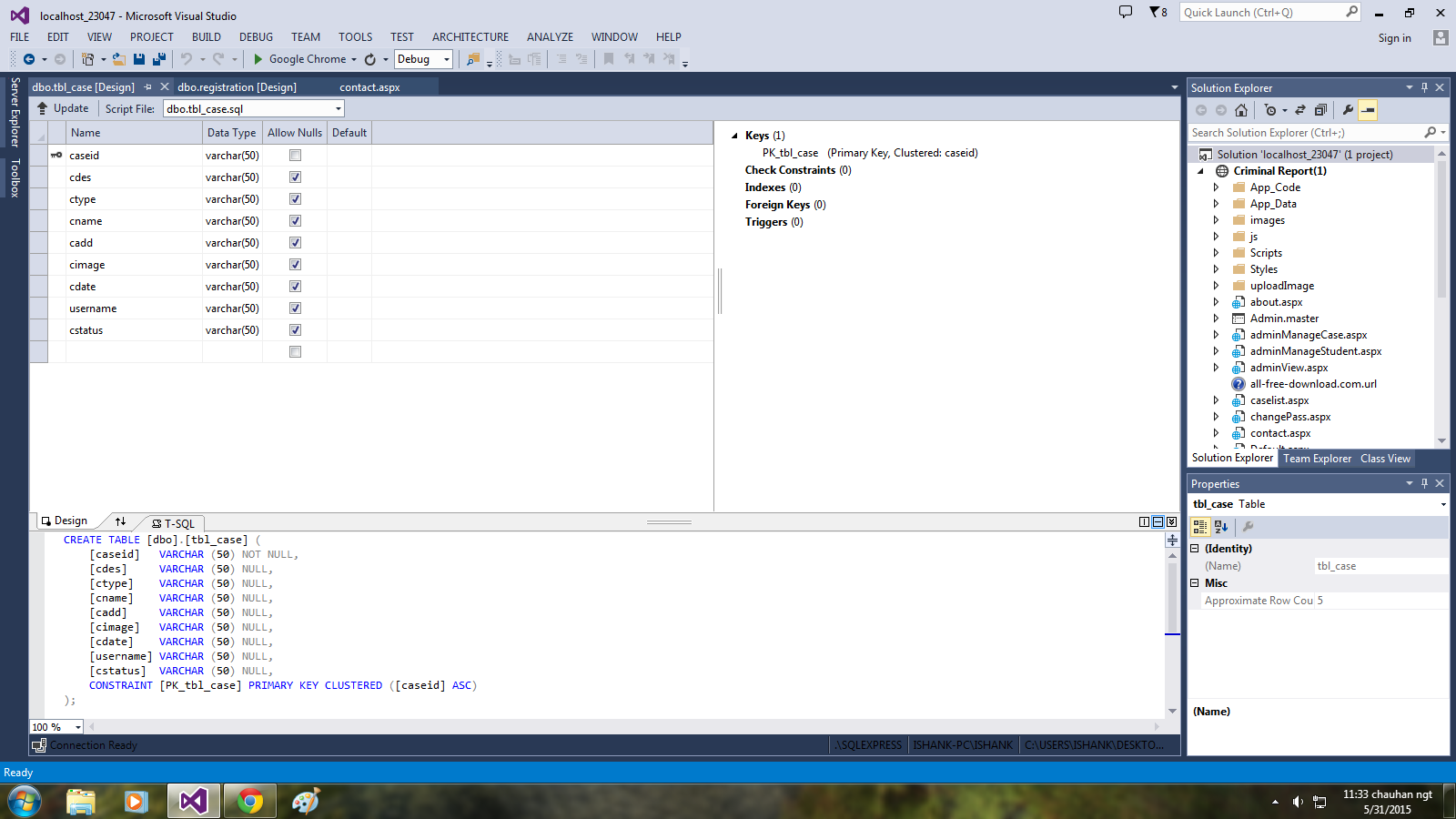
**Tables**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Table Name** | **Description** |
| 1. | Register | Information about registration |
| 2. | Tbbugtttttbbug Case | Information about case |

**Registration**



**Case**

****

**4.5 Input & Output Form**

**Input Design:**

The input design is the link that ties the information system into the user’s world. Input specifications describe the manner in which data enters the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data, or they can result in the production of erroneous information.

**Input Design consists of**

* developing specifications and procedures for data preparation
* Steps necessary to put data into a usable form for processing.
* Data entry, the activity of putting data into the computer processing.

**Five objectives of design input focus on**

* Controlling the amount of input required
* Avoid delay
* Avoiding errors in data
* Avoiding extra steps.
* Keeping the process simple.

Input stages several activities have to be carried out as part of the overall input process. They include some or all of the following.

Data recording (i.e., collection of data)

Data encapsulation (i.e., transfer of data)

Data Conversion (i.e. controlling the flow of data)

Data Transmission (i.e. transporting the data)

Data Validation (i.e. checking the input data)

Data Correction (i.e. correcting the errors)

Input Performa were designed, after a careful discussion with users. It was attempted to cover all user requirements. Designed Performa were given to user for any suggestion and final approval.

Various data items were identified and wherever necessary were recorded.

Input designs are aimed at reducing the chances of mistakes of errors. As the human beings are prone to errors there is always a possibility of occurrence of chance of errors. Adequate validation checks are incorporated to ensure error free data storage. Some of the data validation checks applied are as following:

* Redundancy of data is checked. It means the records of primary key do not occur twice.
* Primary key field of any table must not be left blank.
* Wherever items are coded, input code is checked for it’s validity with respect to several checks.
* Utmost care has been taken to incorporate the validation at each stage of the system. E.g. when entering records into employee information table for employee, it is checked that whether the corresponding employee exists in the employee information table etc.,

**OUTPUT DESIGN**

Output design is one of the most important features of the information system. When the outputs is not of good quality the users will be averse to use the newly designed system and may not use the system. There are many types of output, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used.

Outputs from computer system are required primarily to communicate the results of processing to users, They are also used to provide a permanent hard copy of these results for later consultation. Various types of outputs required can be listed as below:

* External Outputs, whose destination is outside the organization
* Internal outputs, whose destination is with the organization
* Operational outputs, whose use is purely with in the computer department e.g., program-listing etc.
* Interactive outputs, which involve the user is communicating directly with the computer, It is particularly important to consider human factor when designing computer outputs. End user must find outputs easy to use and useful to their jobs, Without quality output, user may find the entire system unnecessary and avoid using it. The term “Output” in any information system may apply to either printer or displayed information. During the designing the output for this system, it was taken into consideration, whether the information to be presented in the form of query of report or to create documents etc.

Other important factors that were taken into consideration are:

The End user, who will use the output.

* The actual usage of the planned information
* The information that is necessary for presentation
* When and how often output and their format is needed. While designing output for project based Attendance Compilation System, the following aspects of outputs designing were taken into consideration.
* The outputs (i.e., well formatted table outputs in the screen itself) designed are simple to read and interpret.
* Format of each output was another important point taken into consideration. Output media, for each output appropriate media is decided whether it will be displayed on screen or will be taken to printer or both.
* Other output design related specifications, i.e., how frequently the outputs will be generated, how many pages or sheets approximately it will keep up, what is its planned use and output distribution to users are also taken into account.

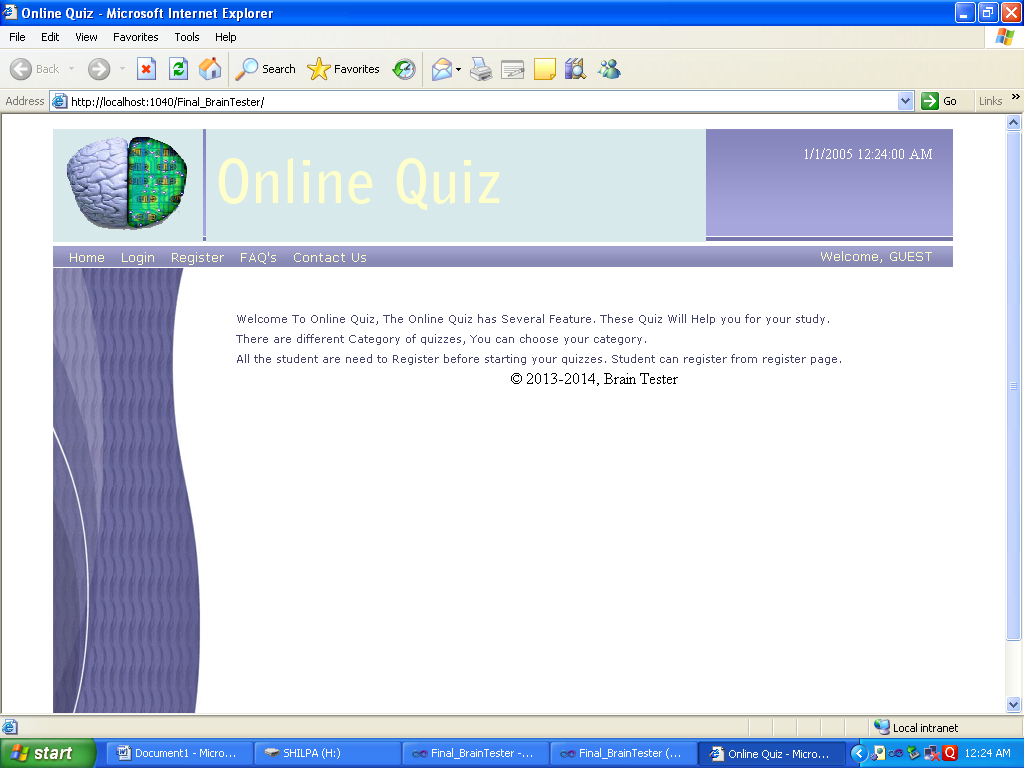
It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

For Example

* + - Will decimal points need to be inserted
    - Should leading zeros be suppressed.

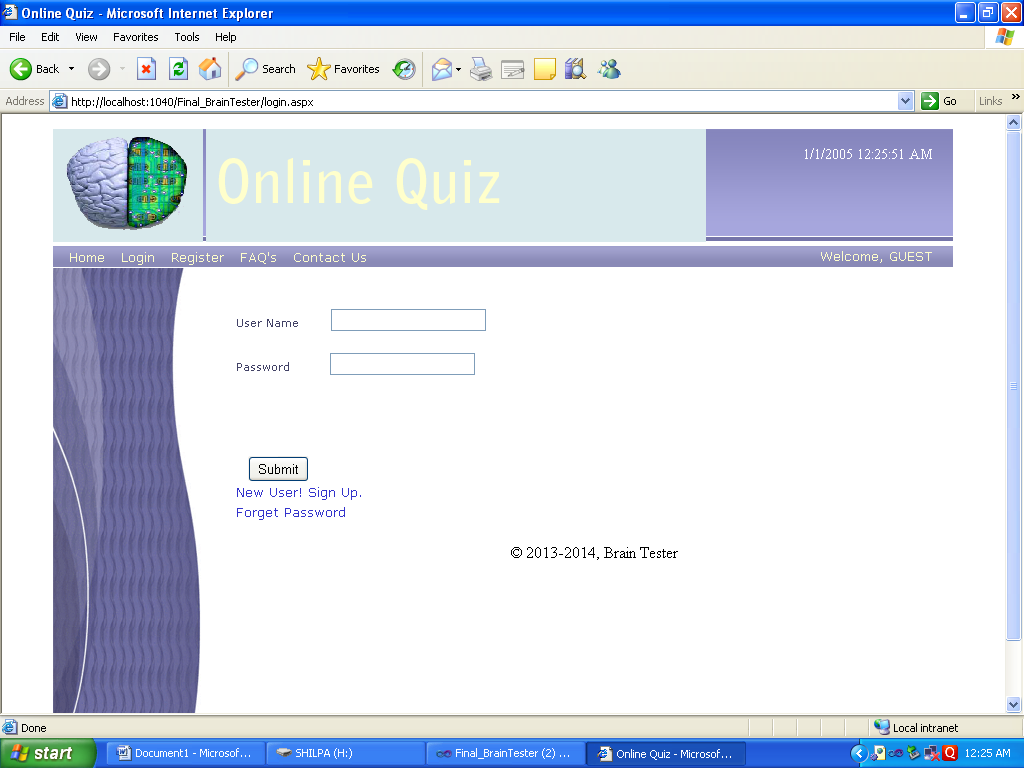
We can see the front page of the web given below:

**Home**



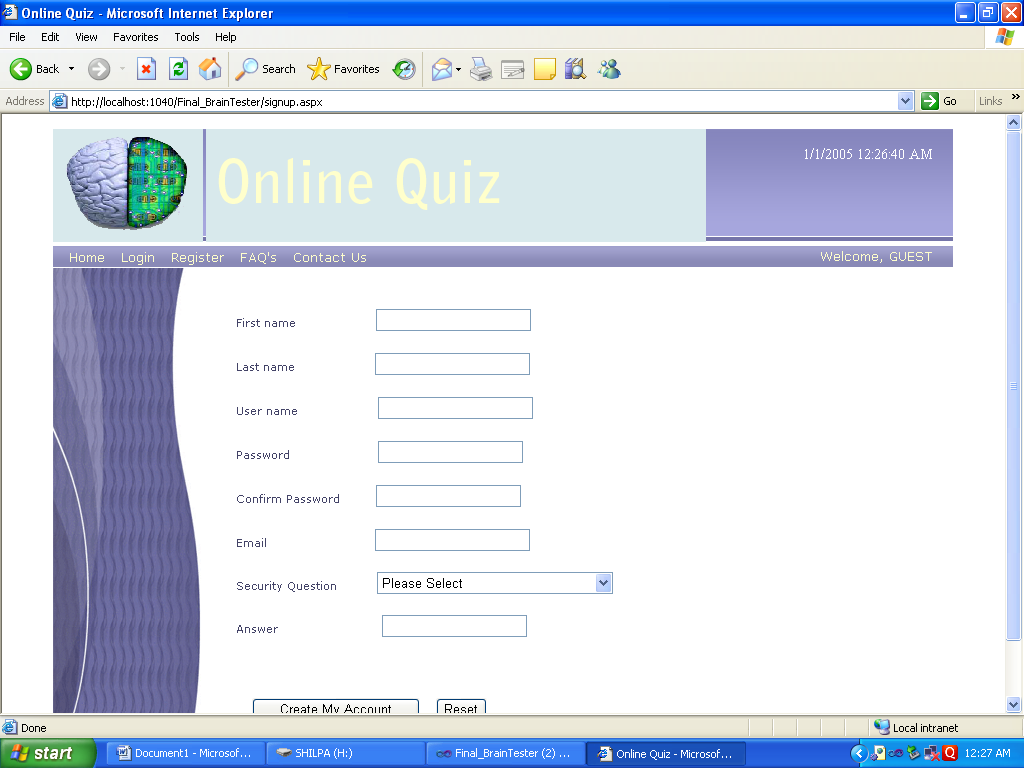
Desc:- This screen shows the detail and purpose of site.

**Log in**



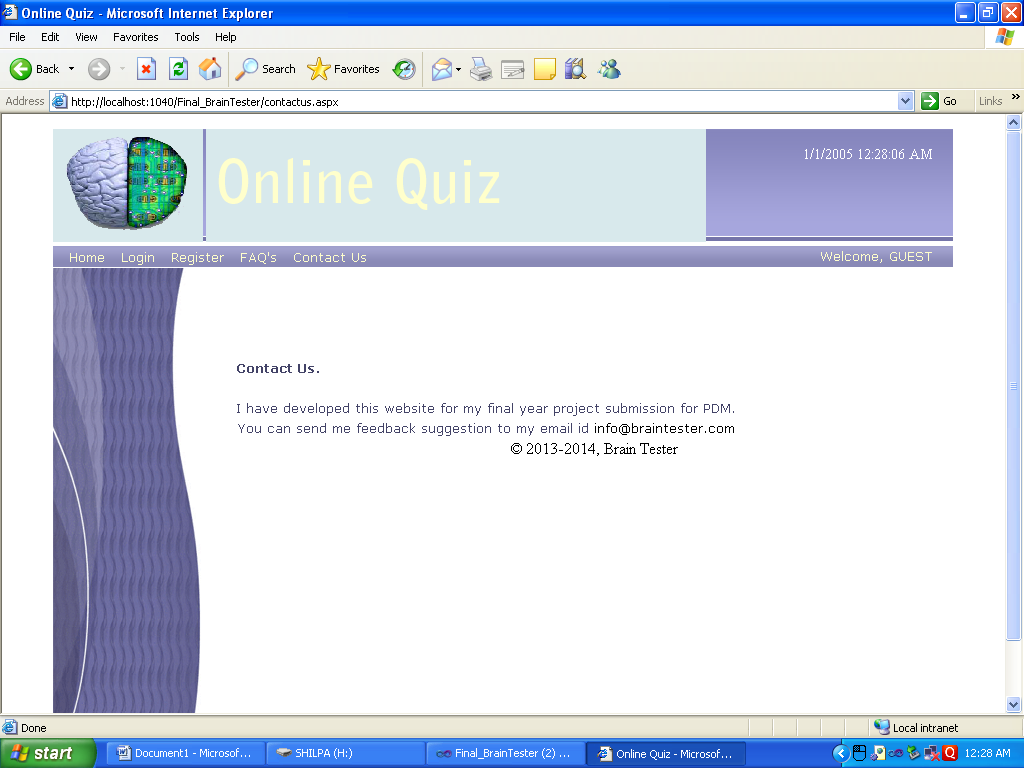
Desc:-This screen shows the controls for login in which user has to fill detailsbefore giving test .

Register



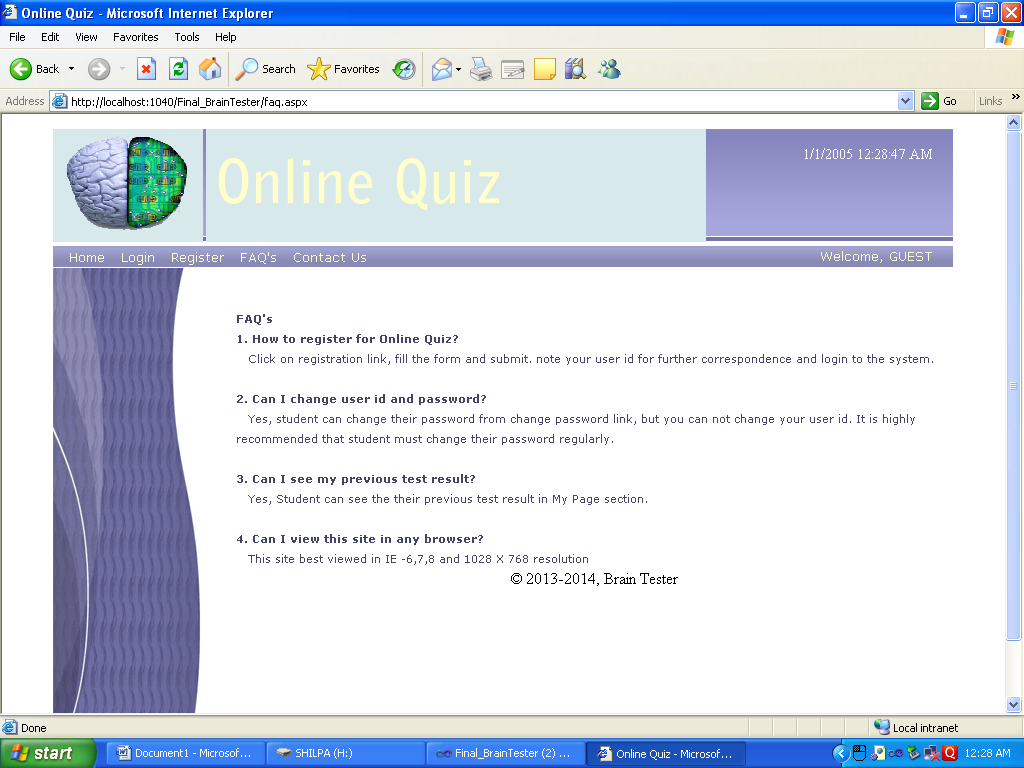
Desc:-this screen shows the required information to get registered.

**Contact**



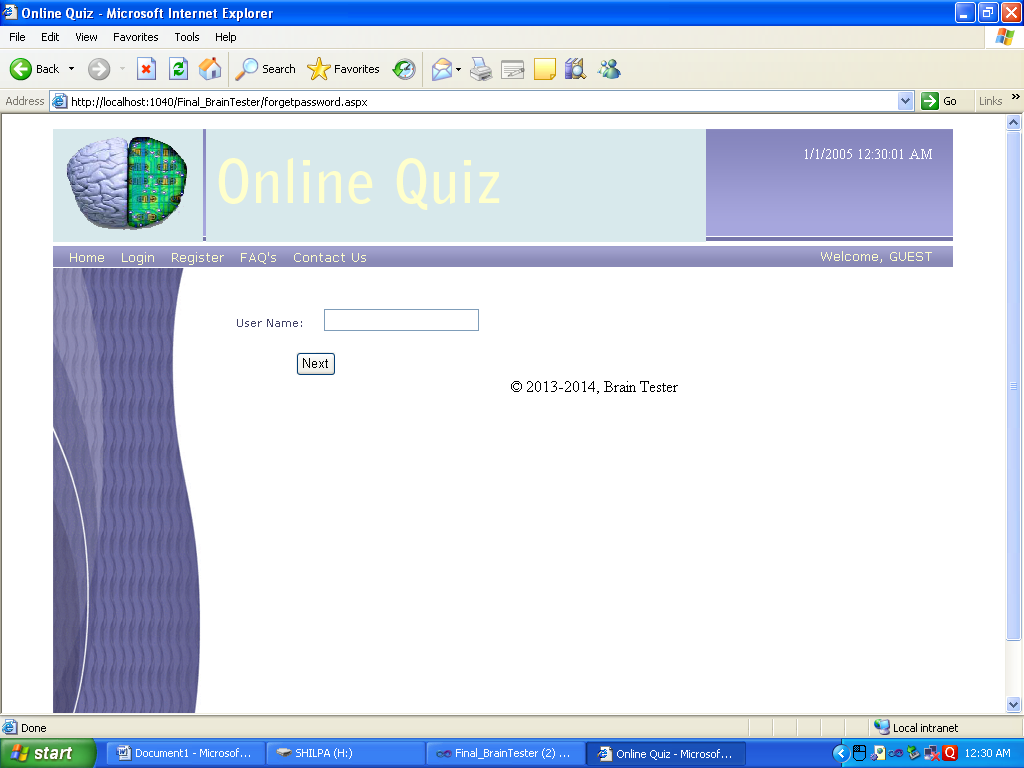
Desc:-This screen shows the information about contact detail.

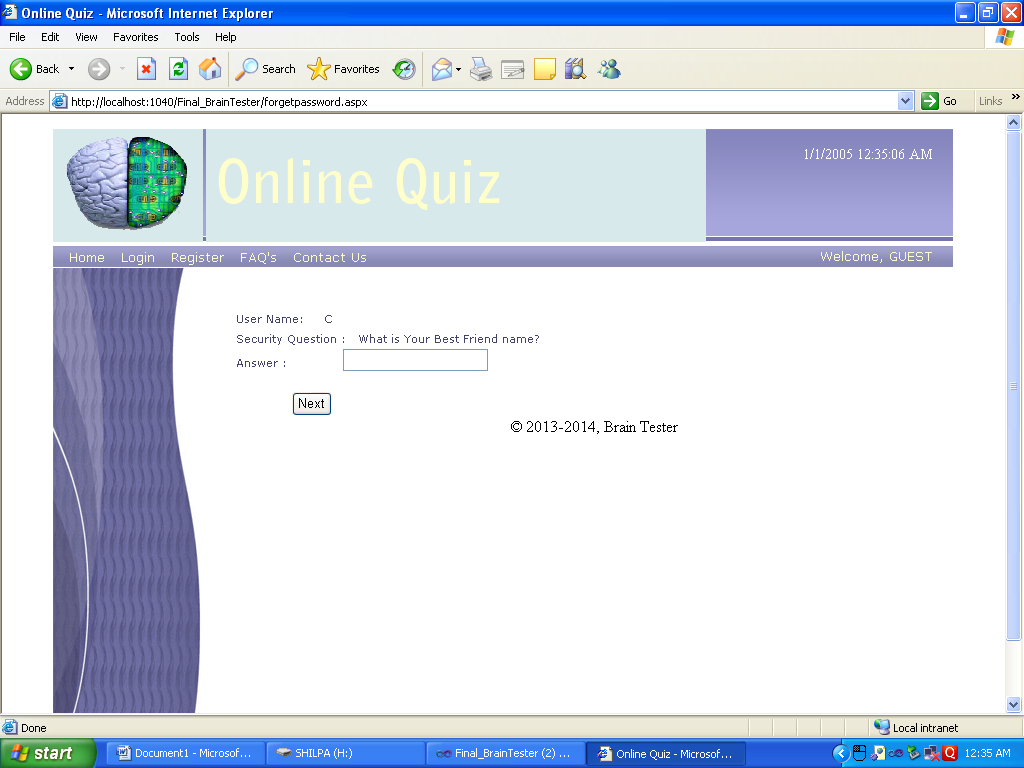
**FAQ’s**

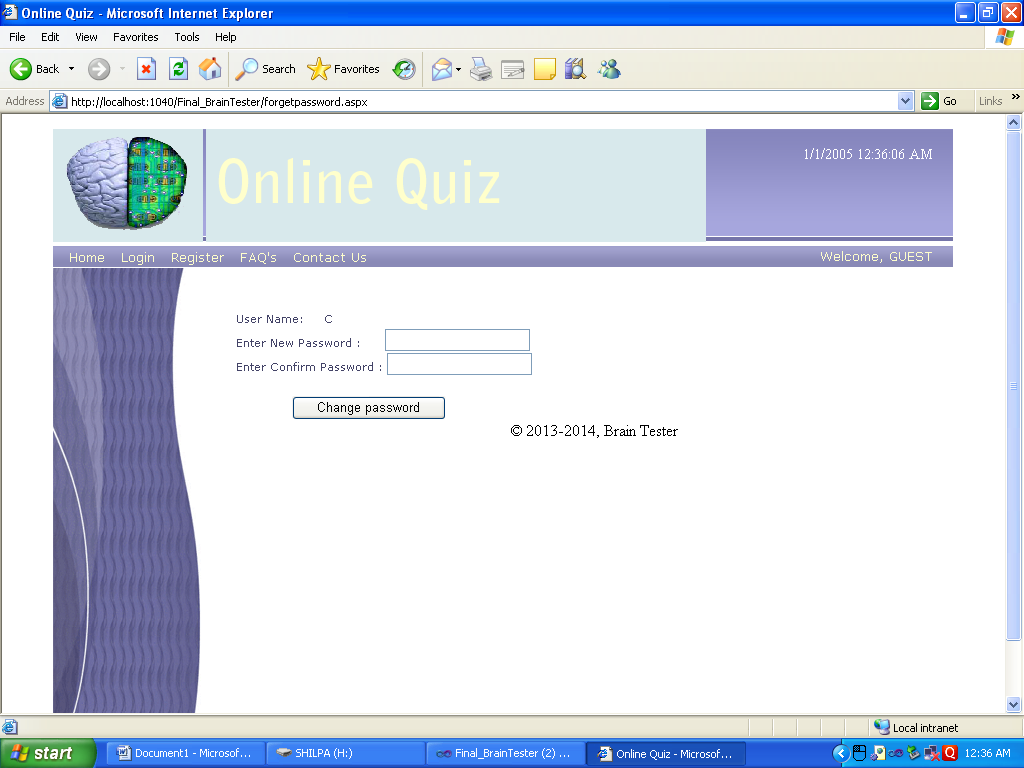


Desc:-This page gives the info about questions that are normaly asked by a user.

**Forget Password**

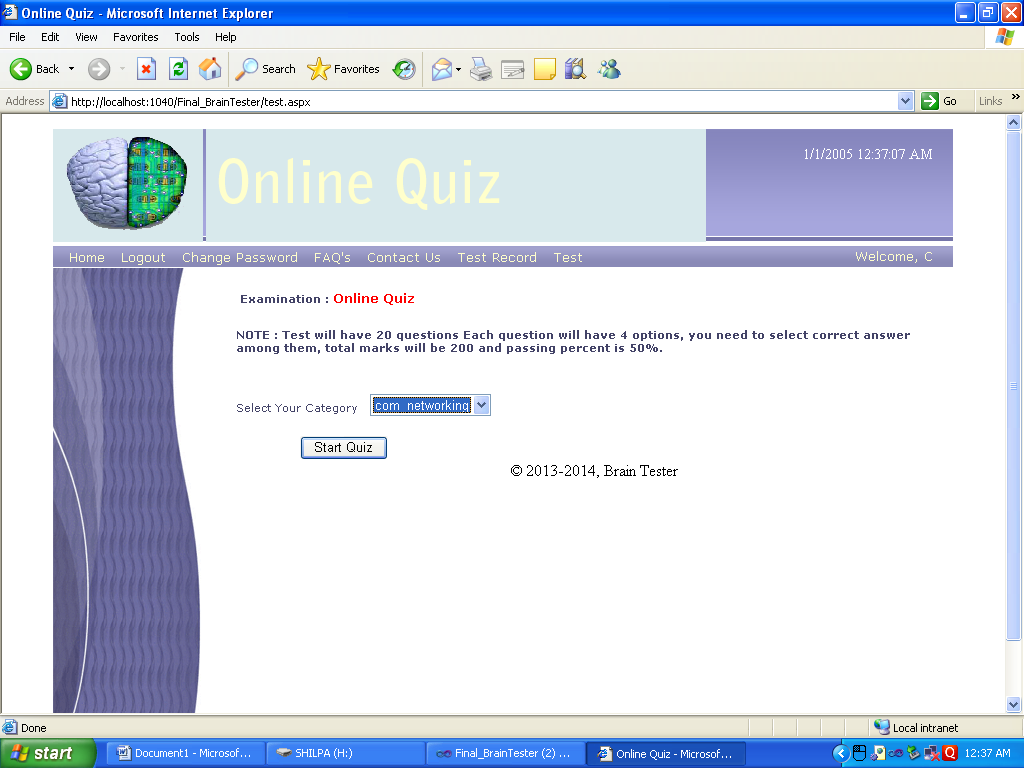






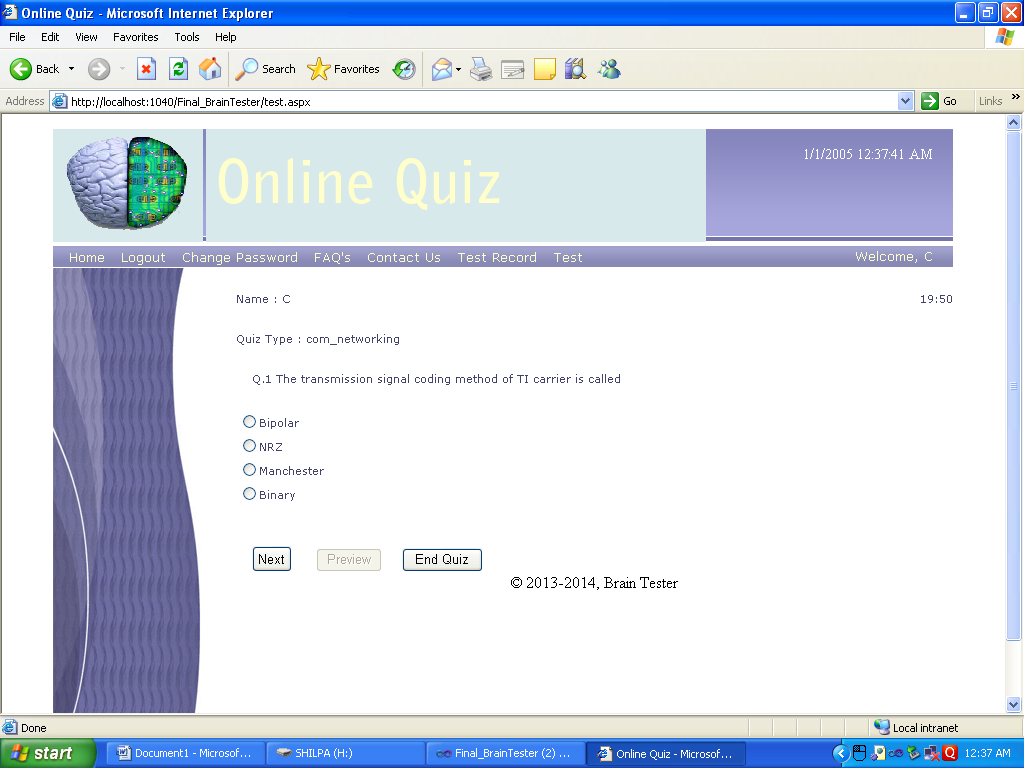
Desc:-These are the password recovery pages which are called sequentially.

**Upload Quiz**



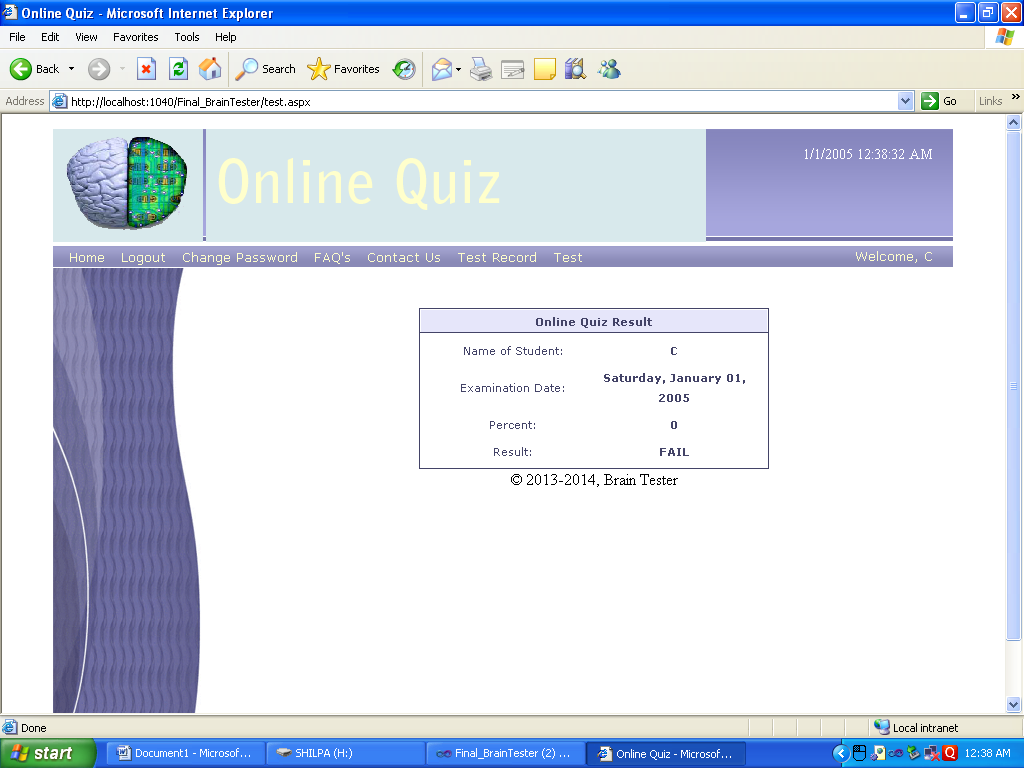
Desc.:This sceen is used by user at the time of test to select the subject.

**Questions**



Desc.:-This screen gives information about the question and options that the user is to select.

**Test Record**



Desc:-This screen shows the test record of the student.

**Chapter 6**

**IMPLEMENTATION / TECHNOLOGICAL ENVIRONMENT**

**IMPLEMENTATION OF THE PROJECT**

Implementation phase of the software development is concerned with translating the design specifications into the source code. After the system has been designed, and arrives the stage of putting it into actual usage known as the implementation of the system. This involves putting up of actual practical usage of the theoretically designed system. The primary goal of implementation is to write the source code and the internal documentation so that conformance of the code to its specifications can easily be verified and so the debugging, modifications and testing are eased. This goal can be achieved by making the source code as clear and as straightforward as possible.

Simplicity, Elegance and Clarity are the hallmarks of good programs whereas complexity are indications of inadequate design and misdirected thinking. The system implementation is a fairly complex and expensive task requiring numerous inter-dependent activities. It involves the effort of a number of groups of people: user and the programmers and the computer operating staff etc. This needs a proper planning to carry out the task successfully. Thus it involves the following activities:

* Writing and testing of programs individually
* Testing the system as a whole using the live data
* Training and Education of the users and supervisory staff

Source code clarity is enhance buy using structured coding techniques, by efficient coding style, by appropriate supporting documents, by efficient internal comments and by features provided in the modern programming language.

The following are the structured coding techniques:

* Single Entry, Single Exit
* Data Encapsulation
* Using recursion for appropriate problems

**INTRODUCTION TO ASP.NET**

Although so Microsoft Visual Basic.NET is a powerful but simple language aimed primarily at developers creating web applications for the Microsoft .NET platform. It inherits many of the best features of C++ and Microsoft Visual Basic, but with some of the inconsistencies and anachronisms removed, resulting in cleaner and logical language. VB also contains a variety of useful new innovations that accelerate application development, especially when used in conjunction with Microsoft Visual Studio .NET.

The Common Language Runtime provides the services that are needed for executing any application that’s developed with one of the .NET languages. This is possible because all of the .NET languages compile to a common Intermediate Language. The CLR also provides the common type system that defines those data types that are used by all the .Net languages. That way, you can use same data types regardless of what.NET language you’re using to develop your application presentations.

**ASP.NET:**  Microsoft, realizing that ASP does possess some significant shortcomings, developed ASP.net. ASP.net is a set of components that provide developers with a framework with which to implement complex functionality. Two of the major improvements of ASP.net over traditional ASP are scalability and availability. ASP.net is scalable in that it provides state services that can be utilized to manage session variables across multiple web services in a server farm. Additionally, ASP.net possesses a high performance process model that can detect application failures and recover from them. We use the fundamentals of programming with VB using Visual Studio .NET and .NET framework.

The project is the starting point for authoring applications, components &services in Visual Studio.NET 2005.It eats as a container that manages your source code, data connections & references. A project is organized as part of a solution, which can contain multiple projects that are independent of each other. C# project file has .asproj extension whereas solution file has .sln extension.

In order to write code against an external component, your project must first contain a reference to it. A reference can be made to the following types of component.

(1) .NET class libraries or assemblies

(2) COM components

(3) Other class libraries of projects in the same solution

(4) XML web services

**FEATURES OF ASP.NET:**

(1) Component Infrastructure.

(2) Language Integration.

(3) Internet Interoperation.

(4) Simple Development.

(5) Simple Deployment.

(6) Reliability.

(7) Security

**Chapter7**

**TESTING AND RESULTS**

**TESTING & RESULTS**

There are a lot of Test Styles are available to test a Software, but I have to discuss few of them which are used in “Clinic Route”. The most important activity at the implementation stage is the system testing with the objective of validating the system against the designed criteria. During the development cycle, user was involved in all the phases that are analysis, design and coding. After each phase the user was asked whether he was satisfied with the output and the desired rectification was done at the moment. During coding, generally bottom up technique is used. Firstly the lower level modules are coded and then they are integrated together.

Thus before implementation, it involves the testing of the system. The testing phase involves testing first of separate parts of the system and then finally of the system as a whole. Each independent module is tested first and then the complete system is tested. This is the most important phase of the system development. The user carries out this testing and test data is also prepared by the user to check for all possible combinations of correct data as well as the wrong data that is trapped by the system. So the testing phase consists of the following:

**Function Testing**

Most test groups start with fairly simple function testing but then switch to a different style, often involving the interaction of several functions, once the program passes the mainstream function tests. Within this approach, a good test focuses on a single function and tests it with middle-of-the road values. We don’t expect the program to fail a test like this, but it will if the algorithm is fundamentally wrong, the build is broken, or a change to some other part of the program has fowled this code.

These tests are highly credible and easy to evaluate but not particularly powerful. Some test groups spend most of their effort on function tests. For them, testing is complete when every item has been thoroughly tested on its own. In my experience, the tougher function tests look like domain tests and have their strengths.

Functional testing of this software was done and it was checked that result generated by the computer through this software and the result generated manually was the same. So the Clinic Route is effectively functioning. There is no biasness in the result generated by this software

**Structural testing:**

Structural testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are required and used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit testing, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

White-box test design techniques include:

* Control flow testing
* Data flow testing
* Branch testing
* Path testing

**TESTING METHODS**

Testing presents an interesting anomaly for the software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test case that are initiated to "demolish" the software that has been build. Infect, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive.

Models of Testing:-

There are different Models of testing. On the basis of testing methods there are two types of testing:

1. Black-box testing.

2. White-box testing

**Black-box tests** are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that integrity of external information is maintained

**White-box tests** are used to examine the procedural details. It checks the logical paths by test case. It can also checks the conditions, loops used in the software coding. It checks that loops are working correctly on defined boundary value.

**WHITE-BOX TESTING**

White box sometimes called “Glass box testing” is a test case design uses the control structure of the procedural design to drive test case.

Using white box testing methods, the following tests where made on the system

a) All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed

b) All logical decisions were checked for the truth and falsity of the values.

**BLACK-BOX TESTING**

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like.

1) Interface errors

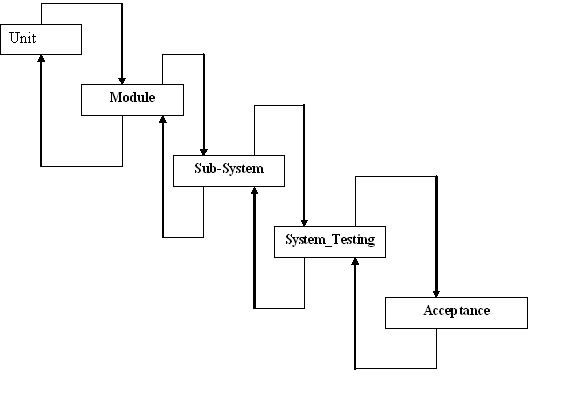
2) Performance in data structure

3) Performance errors

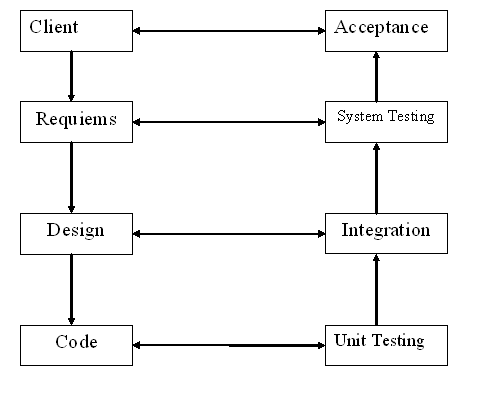
4) Initializing and termination errors

Unlike white-box testing, which is performed earlier in the testing process, black-box testing tends to be applied during later stages of testing. Because black-box testing purposely disregards control structure, attention is focused on the information domain.

**LEVELS OF TESTING**

****

* **UNIT TESTING**
* Unit Testing is primarily carried out by the developers themselves.
* Deals functional correctness and the completeness of individual program units.
* White box testing methods are employed
* **INTEGRATION TESTING**
* Integration Testing: Deals with testing when several program units are integrated.
* **Regression Testing:** Change of behaviour due to modification or addition is called ‘Regression’. Used to bring changes from worst to least.
* **Incremental Integration Testing**: Checks out for bugs which encounter when a module has been integrated to the existing.
* **Smoke Testing**: It is the battery of test which checks the basic functionality of program. If fails then the program is not sent for further testing.
* **SYSTEM TESTING**
* System Testing - Deals with testing the whole program system for its intended purpose.
* **Recovery Testing**: System is forced to fail and is checked out how well the system recovers the failure.
* **Security Testing**: Checks the capability of system to defend itself from hostile attack on programs and data.
* **Load & Stress Testing**: The system is tested for max load and extreme stress points are figured out.
* **Performance Testing:** Used to determine the processing speed.
* **Installation Testing:** Installation & uninstallation is checked out in the target platform.
* **ACCEPTANCE TESTING**
* **UAT** ensures that the project satisfies the customer requirements.
* **Alpha Testing:** It is the test done by the client at the developer’s site.
* **Beta Testing**: This is the test done by the end-users at the client’s site.
* **Term Testing**: Checks out for faults occurrence in a long term usage of the product.
* **Compatibility Testing**: Determines how well the product is substantial to product transition.



**Testing Process**

1. **Test Planning**:-

Describing the scope, approach, resources and schedules of intended test activities, also describe the technical and managerial approach to be followed for testing a component or system.

1. **Test Specification**

Specifies the test inputs, execution condition and predicted result for an item unit, integration and system level test cases are designed on the bases of requirements and as per the plan lay down.

1. **Test Execution:-**

Test cases are executed according to the test procedures prepared for each test designed; test procedure describes the purpose, special requirements and the steps to be followed in executing the test.

1. **Test Recording**:-

All executed test have test log. Whenever discrepancies are observed from the expected result, test incident must be raised. Test log is a record of all relevant details of execution of test. Test incident records any event that occurs during testing that requires investigation.

1. **Test Completion**:-

A test records are checked against check completion criteria. Every test must be completed and must be no defect logged during execution phase that is not fixed by respective developer and if fixed should not remain untested by testing resource.

**CHAPTER 8**

**SYSTEM SECURITIES AND MAINTANENCE**

**8.1 Introduction**

The protection of computer based resources that include hardware, software, data, procedures and people against unauthorized use or natural Disaster is known as **System Security.**

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality

**SYSTEM SECURITY** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

**DATA SECURITY** is the protection of data from loss, disclosure, modification and destruction.

**SYSTEM INTEGRITY** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

**PRIVACY** defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

**CONFIDENTIALITY** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

**Types of System Security**

There are various types of security associated with this system:

1. **Physical Security:-**Damage due to natural causes like earth tremor, flooding, water logging, fire hazards, atmospheric or environmental conditions etc. For overcoming these difficulties the replica of the data are automatically stored at various networks and for environmental conditions Air conditioning environment is created.
2. **Data security:-**

There are basically two problems associated with data security:-

1. Data not being available to the authorized person at the time of need.
2. Data becoming available to the unauthorized person.

To overcome these difficulties the following access facilities has been provided:-

1. ***Identification: -***Unique Ids for the different users have been provided.
2. ***Authentication:-***System checks the password under the particular user identification. The computer permits the various resources to the authorized person.
3. ***Authorization:*** The access control mechanism to prevent unauthorized logging to the system.

**3. Web server Security**

There is a wide range of very flexible security features one can implement on ones’ Web server. Here’s a summary:

* Access to Web servers, individual Web pages, and entire directories containing Web pages can be set to require a username and password.
* Access to Web servers, individual Web pages, and entire directories containing Web pages can be limited to customers on specific computer systems. (In other words, access will be denied unless the user is at his or her usual computer or workstation.)

**4. Secure/ Encrypted Transactions**

One can further enhanced security on ones’ intranet by encrypting Web transactions. When one use an encryption facility, information submitted by customers using Web fill-in forms-including usernames, passwords, and other confidential information-can be transmitted securely to and from the Web server.

It’s ones’ responsibility to determine the level of security one need on ones’ intranet/system, and, of course, to implement it. Putting most of the security measures mentioned into place is not difficult. Ones’ primary concern will be explaining to customers how intranet security works, not so much as a limiting factor but as an opportunity for increased use and collaboration using ones’ intranet. Assuring decision-makers that they can make information available on ones’ intranet in a secure fashion can go a long way toward making ones’ intranet a success. At the same time, it’s important to make sure both information providers and their customers understand a number of critical aspects of intranet security, so they don’t inadvertently defeat the purpose of it.

**8.2 System Maintenance**

**Definition of Maintenance**

The use of the word maintenance to describe activities undertaken on software systems after delivery has been considered a misnomer due to its failure to capture the evolutionary tendency of software products. Maintenance has traditionally meant the upkeep of an artifact in response to the gradual deterioration of parts due to extended use, which is simply corrective maintenance. So for example, one carries out maintenance on a car or a house usually to correct problems e.g. replacing the brakes or fixing the leaking roof. If however we were to build an extension to the house or fit a sunroof to a car then those would usually be thought of as improvements (rather than a maintenance activity). Therefore to apply the traditional definition of maintenance in the context of software means that software maintenance is only concerned with correcting errors. However, correcting errors accounts for only part of the maintenance effort.

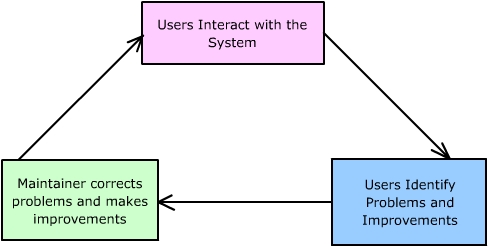
Consequently, a number of authors have advanced alternative terms that are considered to be more inclusive and encompass most, if not all, of the activities undertaken on existing software to keep it operational and acceptable to the users. These include “software evolution”, “post-delivery evolution” and “support”. However it can be argued that there is nothing wrong with using the word maintenance provided software engineers are educated to accept its meaning within the software engineering context regardless of what it means in non-software engineering disciplines. After all, any work that needs to be done to keep a software system at a level considered useful to its users will still have to be carried out regardless of the name it is given.

As a result of the above, attempts have been made to develop a more comprehensive definition of maintenance, which would be appropriate for use within the context of software systems. Some definitions focus on the particular activities carried out during maintenance, for example, According to Martin and McClure (1983), software maintenance must be performed in order to

* Correct errors
* Correct design flaws
* Interface with other systems
* Make enhancements
* Make necessary changes to the system
* Make changes in files or databases
* Improve the design
* Convert programs so that different hardware, software, system features, and telecommunications facilities can be used

**Why is Software Maintenance Necessary?**

In order to answer this question we need to consider what happens when the system is delivered to the users. The users operate the system and may find things wrong with it, or identify things they would like to see added to it. Via management feedback the maintainer makes the approved corrections or improvements and the improved system is delivered to the users. The cycle then repeats itself, thus perpetuating the loop of maintenance and extending the life of the product. In most cases the maintenance phase ends up being the longest process of the entire life cycle, and so far outweighs the development phase in terms of time and cost. Error! Reference source not found. Shows the lifecycle of maintenance on a software product and why (theoretically) it may be never ending.



**Figure 7.1**

**Types of Software Maintenance**

In order for a software system to remain useful in its environment it may be necessary to carry out a wide range of maintenance activities upon it.

**1. Corrective:-**

Changes necessitated by actual errors (defects or residual "bugs") in a system are termed corrective maintenance. These defects manifest themselves when the system does not operate as it was designed or advertised to do.

A defect or “bug” can result from design errors, logic errors and coding errors. Design errors occur when for example changes made to the software are incorrect, incomplete, wrongly communicated or the change request misunderstood. Logic errors result from invalid tests and conclusions, incorrect implementation of design specification, faulty logic flow or incomplete test data. Coding errors are caused by incorrect implementation of detailed logic design and incorrect use of the source code logic. Defects are also caused by data processing errors and system performance errors. All these errors, sometimes called “residual errors” or “bugs” prevent the software from conforming to its agreed specification.

In the event of a system failure due to an error, actions are taken to restore operation of the software system. The approach here is to locate the original specifications in order to determine what the system was originally designed to do. However, due to pressure from management, maintenance personnel sometimes resort to emergency fixes known as “patching”. The ad hoc nature of this approach often gives rise to a range of problems that include increased program complexity and unforeseen ripple effects.

Corrective maintenance has been estimated to account for 20% of all maintenance activities.

**2. Adaptive:-**

Any effort that is initiated as a result of changes in the environment in which a software system must operate is termed adaptive change. Adaptive change is a change driven by the need to accommodate modifications in the environment of the software system, without which the system would become increasingly less useful until it became obsolete.

The term environment in this context refers to all the conditions and influences which act from outside upon the system, for example business rules, government policies, work patterns, software and hardware operating platforms. A change to the whole or part of this environment will warrant a corresponding modification of the software.

Unfortunately, with this type of maintenance the user does not see a direct change in the operation of the system, but the software maintainer must expend resources to effect the change. This task is estimated to consume about 25% of the total maintenance activity.

**3. Perfective:-**

The third widely accepted task is that of perfective maintenance. This is actually the most common type of maintenance encompassing enhancements both to the function and the efficiency of the code and includes all changes, insertions, deletions, modifications, extensions, and enhancements made to a system to meet the evolving and/or expanding needs of the user. A successful piece of software tends to be subjected to a succession of changes resulting in an increase in its requirements

As the program continues to grow with each enhancement the system evolves from an average-sized program of average maintainability to a very large program that offers great resistance to modification. Perfective maintenance is by far the largest consumer of maintenance resources; estimates of around 50% are not uncommon.

**4. Preventive:-**

Preventative maintenance is "maintenance performed for the purpose of preventing problems before they occur". This is the process of changing software to improve its future maintainability or to provide a better basis for future enhancements.

The preventative change is usually initiated from within the maintenance organization with the intention of making programs easier to understand and hence

facilitate future maintenance work. Preventive change does not usually give rise to a substantial increase in the baseline functionality.

Preventive maintenance is rare (only about 5%) the reason being that other pressures tend to push it to the end of the queue. For instance, a demand may come to develop a new system that will improve the organization’s competitiveness in the market. This will likely be seen as more desirable than spending time and money on a project that delivers no new function. Still, it is easy to see that if one considers the probability of a software unit needing change and the time pressures that are often present when the change is requested, it makes a lot of sense to anticipate change and to prepare accordingly

**CHAPTER 9**

**CONCLUSION AND FUTURE ENHANCEMENT**

The system has been developed for the given condition and is found working effectively. The developed system is flexible and changes can be made easily whenever required. Using the facilities and functionalities of .Net, the software has been developed in a neat and simple manner, thereby reducing the operator’s work.

The speed and accuracy are maintained in proper way. The user-friendly nature of this software developed in .Net framework is very easy to work with both the higher management as well as other users with little knowledge of computer. The results obtained were fully satisfactory from the user point of view.  
The system was verified with valid as well as invalid data in each manner. The system is run with an insight into the necessary modifications that may be required in the future. Hence the system can be maintained successfully.

**FUTURE EXTENSIONS**

* It is possible to create our own mail server.
* It is possible to create chat server so that user can communicate with each other.
* It is possible to provide searching facility .
* It is possible to registered cases in the system.

**CHAPTER 10**

**BIBLIOGRAPHY**

MASTERING IN ASP.NET

Visual Studio .NET ……….……………..……………………..Black

MS Press - Microsoft Visual ASP .NET Step By Step

Visual Studio .NET - (O'Reilly) Programming ASP.NET

Visual Studio .NET developers guide by Wrox.

Visual Studio .NET Introduction to ASP.NET unleased.

McGraw Hill-- SQL - The Complete Reference

SQL Server The Complete Reference…………………….George Koth Kevin Loney