Project Report

On

"ONLINE - EXAMINATION"

By
Manish Pant
www.manish.pro

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

***** CONTENTS *****

1 Project proposal Report
2 INTODUCTION.
3 OBJECTIVES OF THE PROJECT
4 SYSTEM ANALYSIS
5 MODULES
6 SYSTEM REQUIREMENT SPECIFICATION
❖ HARDWARE SPECIFICATIONS
❖ SOFTWARE SPECIFICATIONS
7 SYSTEM FLOW CHART
8 DATA FLOW DIAGRAM
9 DATA DICTIONARY
10 ENTITY RELATIONSHIP MODEL
11 E-R DIAGRAM
12 TESTING
❖ UNIT TESTING
❖ INTEGRATION TESTING
❖ VALIDATION TESTING
❖ SYSTEM TESTING
❖ SECURITY TESTING
12 FUTURE SCOPE

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

INTRODUCTION

Online Examination System (OLES) is a website in which the many students simultaneously appeared in the exam. The exam can be of different technology and the question paper will be randomly generated. The evolution of all the papers is done online by the softwareThe Integrated Software Solutions include complete solutions in Enterprise Resource Planning, Customer Relationship Management, Executive Information Systems, Supply Chain Management, e-Business solutions, Development of Application Software and customized solutions, IT Facility Management and other related professional services. The Solutions we provide are complemented by full range of services, creating a single window for customers for all their IT needs, thus facilitating single point of accountability.

Objective:

Online Examination System, System will hold ONLINE tests in different technology to candidates and evaluate their scores in various tests and store the results of registered candidates after that provided certified professional certificate. Further, He or She who is searching Jobs in MNC This score which have got in this Test, Help for getting job and The all Companies are given the values of this scores So the user will be ahead to all the competitors and user will be get benefit in the company's Interview .

SYSTEM ANALYSIS

Systems Development can generally be thought of as having two major components: Systems Analysis and Systems Design. Systems Design is the process of planning a new business system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used (if at all) to make its operation more effective. Systems analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system.

Current System Summary

FEASIBILITY STUDY:

It means to carry out the detailed study of the existing system, find out the problem related to technical, operational staff, and economical field like cost by concerning the user of the study proposal of the proposed system (i.e. creditor ledger) is accepted by the management it will lead to the investigation of the existing system or problem area. Having recognized the problem, the next step to carry overall analysis of the system requirement in terms of it's:

- The input data
- Type of processing needed
- The output

Three phases of the feasibility study is used

Technical feasibility:

It is concerned with the available hardware and the software resources whether they meet the given requirement of the analyzed system or not which include latest machinery and the technique required handling the system. It may also invoke the study of the new alternative to solve the given problem.

Behavioral Feasibility:

People are inherently resistant to change, and computers have known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover, transfers, restraining, and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of conducting business.

Economical feasibility-

It deals with the study of the cost benefit analysis. All the cost of the new system compared with the benefits, which can be obtained for management approval. The benefit may be quantities in nature Current System Summary. The genuine consideration of the system being developed is the approach we follow to look the system in the way it is useful for us.

Modules:

OLSE Project includes the following modules:

1) User
2) Registration
3) Administrator
4) Test
5) Database Design

1) **User & Registration**: User module is related with the users of the OLSE system.

- There are three types of users of the system, users who wants to take test, admin of the system, and guest users for demo.
- A guest user can only take sample tests, and access certain areas of site.
- A guest user can register oneself with the site and avail the facilities provided by the site.
- A user can register oneself under different membership types. Membership types available are:
- Platinum
- Golden
- Silver
- Administrator can log into the site. He/she will have complete control and access to

Each part of the site
- There will be a payment plan for the users of the site (quarterly, annually, Semi – annually).

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

- The payment will be based on membership types.
- If a user applies for certification, the payments will differ.
- Payment mode for any type of payment will be credit/debit card.
- Certificates will be delivered to the user after payment has been made.
- User can ask queries.
- User can fill the feed back form
- A registered user will login into the system.
- A session of the registered user will be maintained
- After logging in to the system, a user can view his/her profile.
- A registered can view all the tests.
- A registered user can apply for a test.
- A registered user can change his/her personal details.
- A registered user can view public transcripts.
- A registered user can view his/her private transcripts.
- A registered user can view scores in a particular test.
- A registered user can view overall performance.
- An administrator will be displayed admin related tasks.

2) Administrator

- Admin module relates to administrative tasks.
- It also deals with admin login.
- The admin can view all the information related to site after logging in.
- He can add new tests.
- He can grant and revoke privileges to certain users of site.
- Admin will enjoy all the privileges related to an admin.

3) Test

- A registered user who wants to take test applies for test.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

- He/she can take tests as per his/her wish.
- A registered user can select various subjects and their topics on which he/she wants to test himself/herself.
- Their scores will be calculated and displayed to them
- He/She can fill the feed back form.
- These results will be stored in the database.

4) **Database Design**

- This module is related with identification of tables that is to be used in the project.
- The project leader with software designers will be responsible for identification of

  Tables, their attributes and relationship amongst them.

Report Generation:

Any project is developed for getting the desired output from the system. It is very important for the user to enter accurate information so as to get accurate output. To help this, this system is bringing designed to how many fields. This is not only made it easier for user to use the system but also reduce the mistakes made by them.

Reports are the most important output of any system. This help is not only in the monitoring the day to day active ties in terms of transactions, user login, results etc but also create MIS reports and help for creating future plans.

Results:

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

These reports are creating by the administrator use with the User, Question & answer table, test table. These reports give the description of the user results after giving the online test.

Merit list:

These reports are created by the administrator use Result Database. These reports give the description of the merit list of the user who had given the test. Top 20 students are listed in this list.

Certification:

This report tells about the certification charge, date and the certification type it's also create by the administrator with the help of user and result database.

Reason for Choosing Asp.Net as a Front End

Introduction the .NET Platform:

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. The runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

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

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

Reasons for Choosing SQL 2005 as Back End

A database system is basically a computer based record keeping system. We have chosen sql server as Database System because it is easily available as it comes part of sql suite.

The Sql Server 2005 relational database engine supports the feature required to support demanding data processing environments .The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the databases.

System RequirementS->

System requirement is two types.

1 **External Interfaces:**
2

1. User Interfaces

This is a web based project so the user interfaces will typically be of web – based type. It is based Visual Studio .NET framework because it is open source, flexible, development is easy and fast.

3 **Hardware Interfaces**

It's a web – based project, so a robust hardware configuration is required. The hardware requirements are:

i) Processor	:	PIV and above.
ii) Motherboard	:	Intel 845 and above
iii) RAM	:	256 to 768 MB
iv) Hard Disk	:	2.5 GB for Visual Studio .NET and 1 GB for Microsoft SQLServer2000.
v) Network Card	:	Standard Ethernet card for networking.
vi) I/O Devices	:	Keyboard, mouse and Color monitor

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

vii) Wires : Twisted pair for networking.

II) Internal Interfaces:

1. Software Interfaces

Following software are required for developing OLSE application:

i) Operating System	: Windows NT, XP etc.
ii) Platform	: ASP.NET, C#.
iii) Environment	: VisualStudio.NET 1.1.
iv) Versioning Tools	: IIS6.0
v) RDBMS	: SqlServer2000

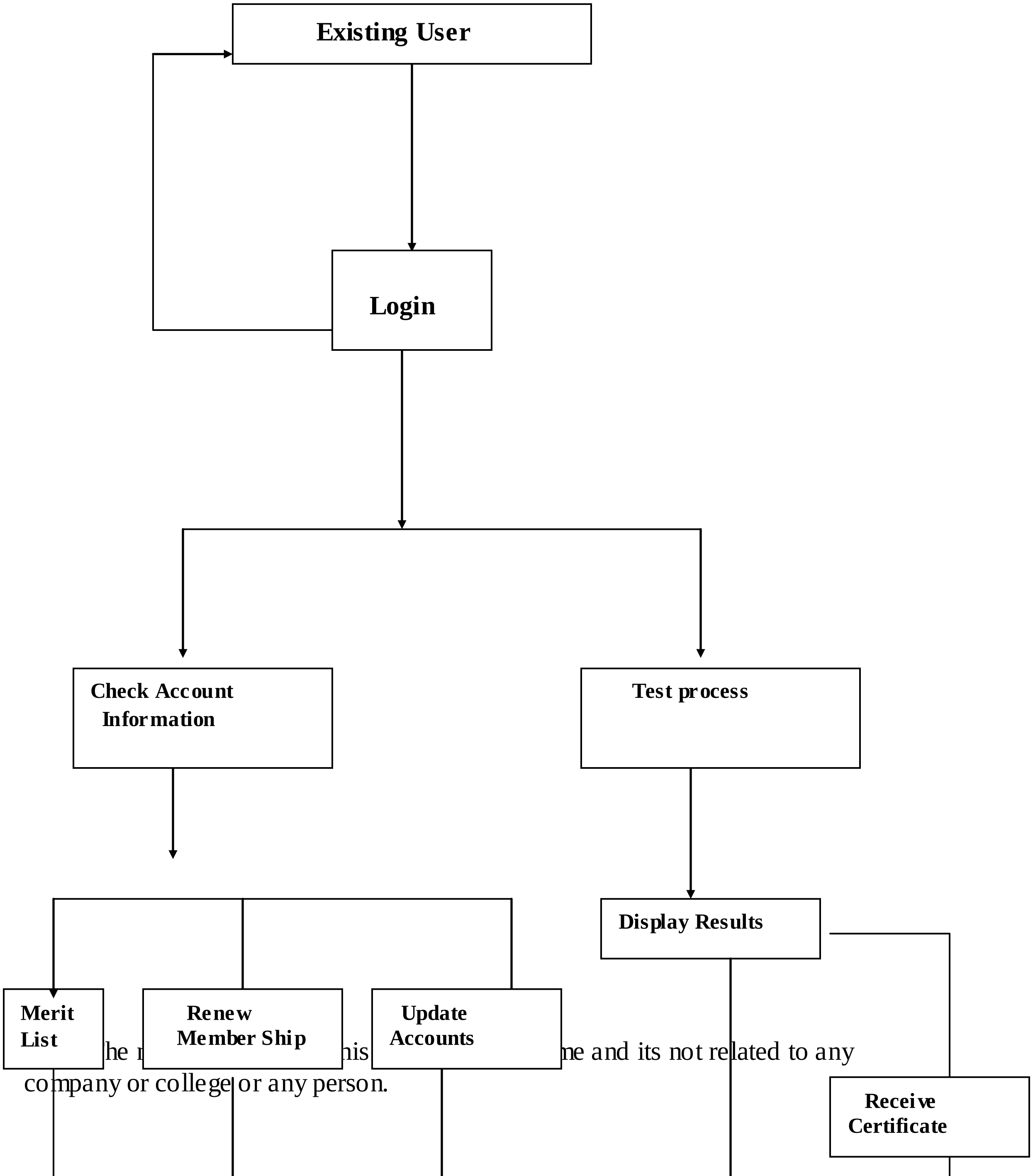
Software Category:

This software is related to the database and web based and also using the concept of Object Oriented Language through Visual C# language. So that Software Category is That "RDBMS/OOP".

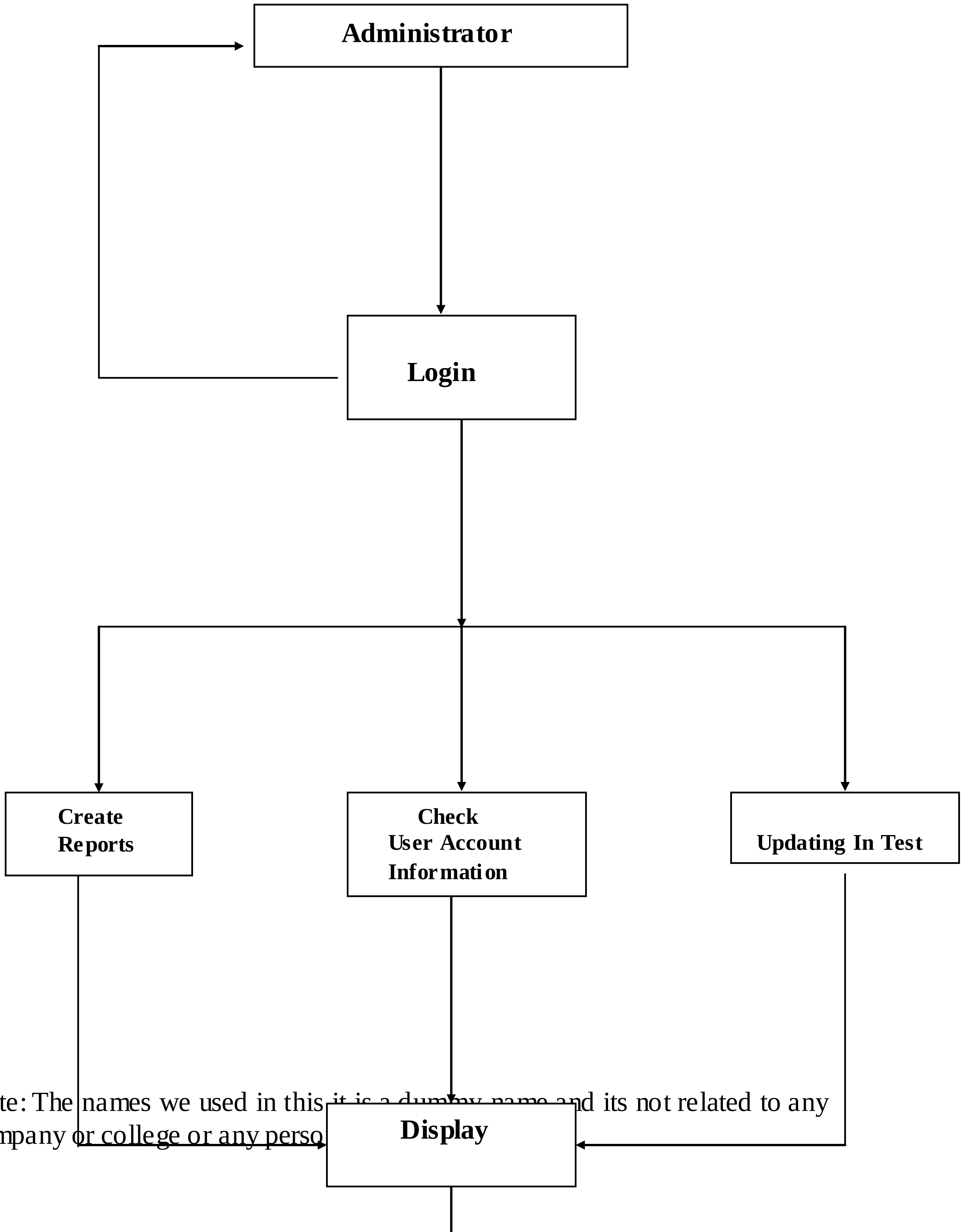
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

1. System Flow Chart

Process logic for user module:

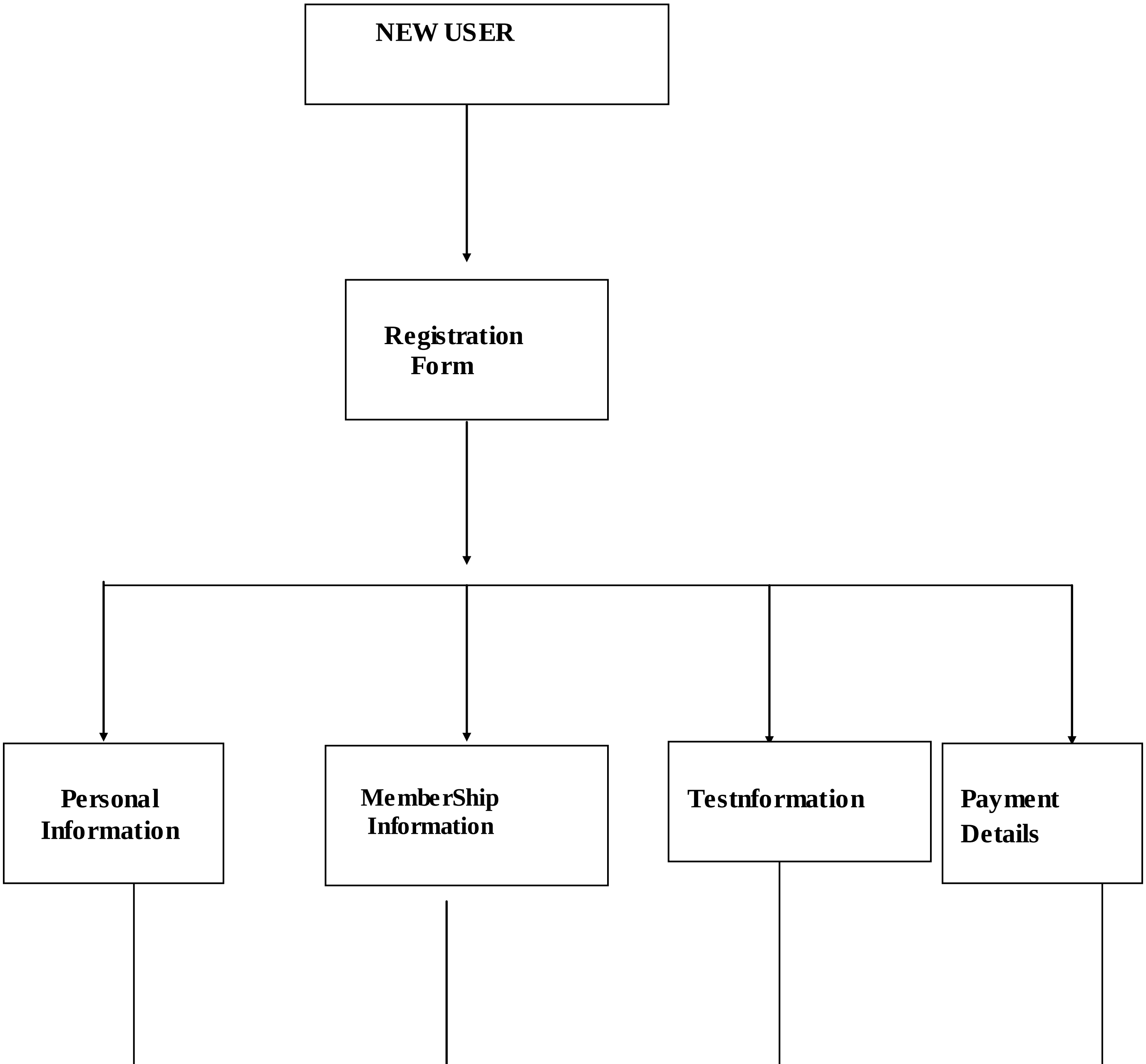


Process logic for Administrator module:



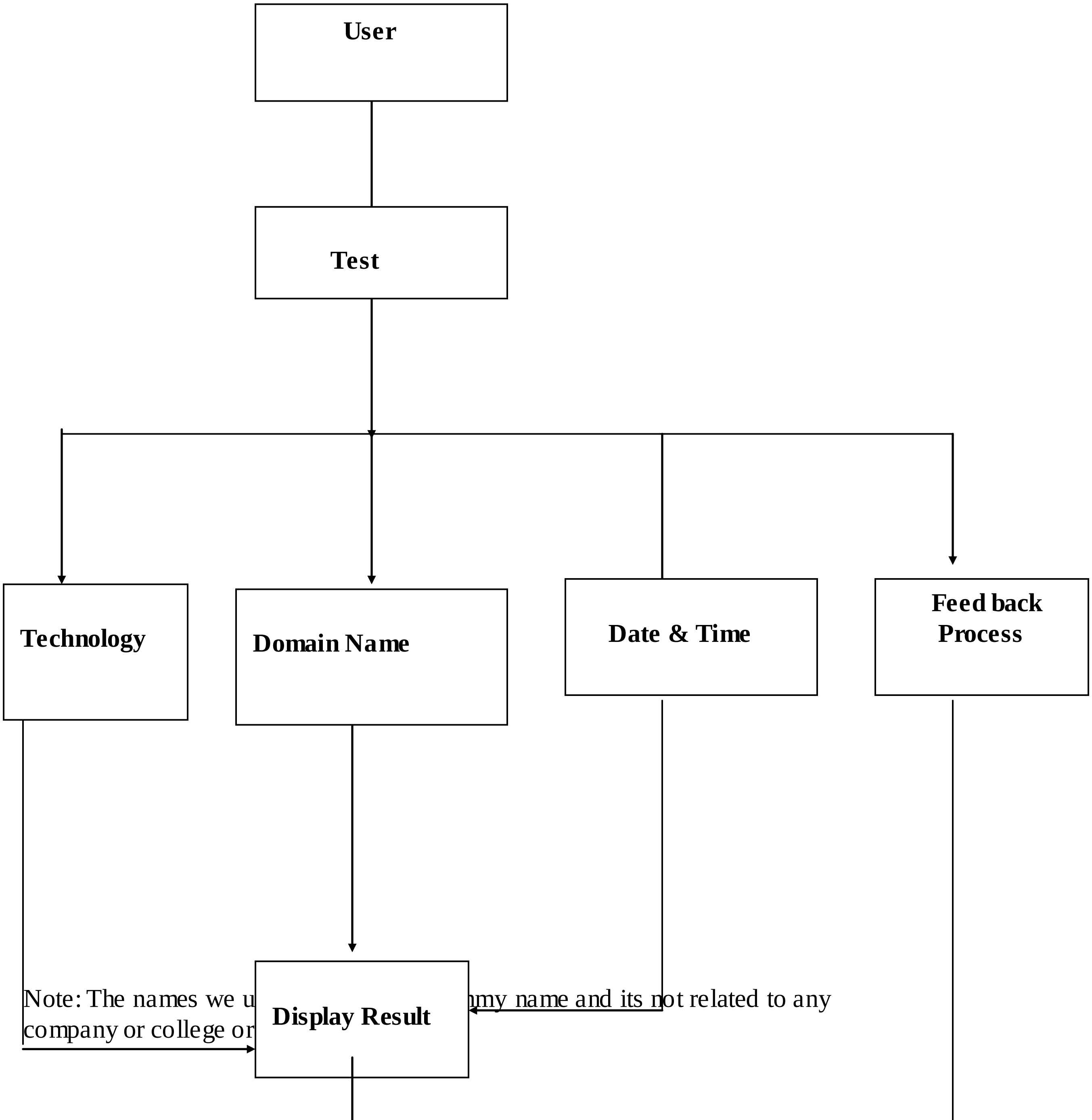
Note: The names we used in this it is a dummy name and its not related to any company or college or any perso

Process logic for Registration module:



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Process logic for Test module:



Note: The names we u my name and its not related to any
company or college or

Data flow diagram

The Data Flow Diagram (DFD.) was first developed by 'Larry Constantine' as a way of expressing system requirements in a graphical form. D.F.D. represents "what data flow rather than how they are processed".

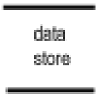
DFD knows as "bubble chart", has a purpose of clarifying system requirements and identifying major transformation that will become the programs in system designs.

A DFD. consists of a series of bubbles joined by lines. The bubble represents data transformation & line represents the data flow in the system.

Basic Data Flow Notation

1) Rectangle (external entity) - a producer or consumer of information that resides outside the bounds of the system to be modeled	□
2) Circle (process) - a transformer of information that resides within the bounds of the system to be modeled	○

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

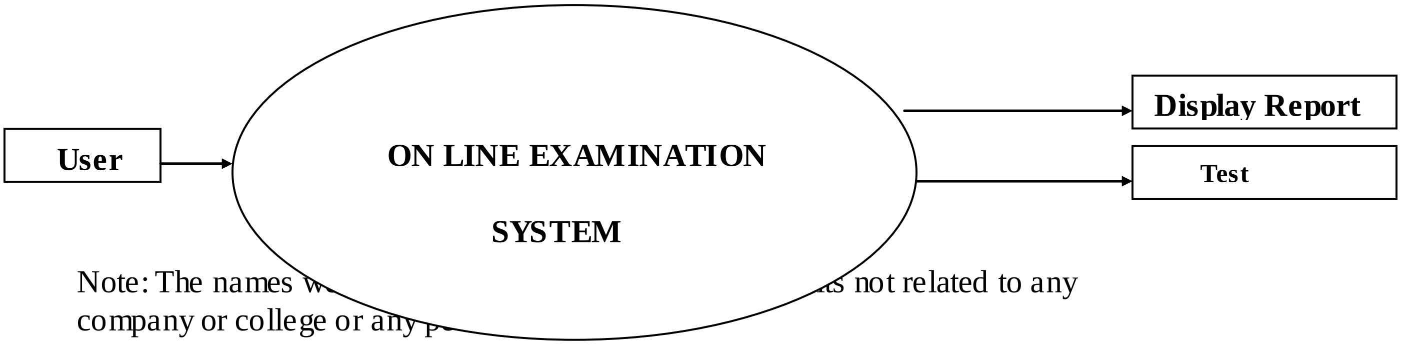
3) Line with an arrow (data item) - a single item, or a collection of data items. Arrow head represents the direction of the data	→
4) Two parallel lines (data store) - a repository of data that is to be stored for use by one or more processes; may be as simple as a buffer or a queue or as sophisticated as a relational data base	

CONTEXT DIAGRAM

Diagram plays a very important role in studying the current system .It contain a single process which the system interface to the library or can say it determines the boundaries. It is a top level DFD.

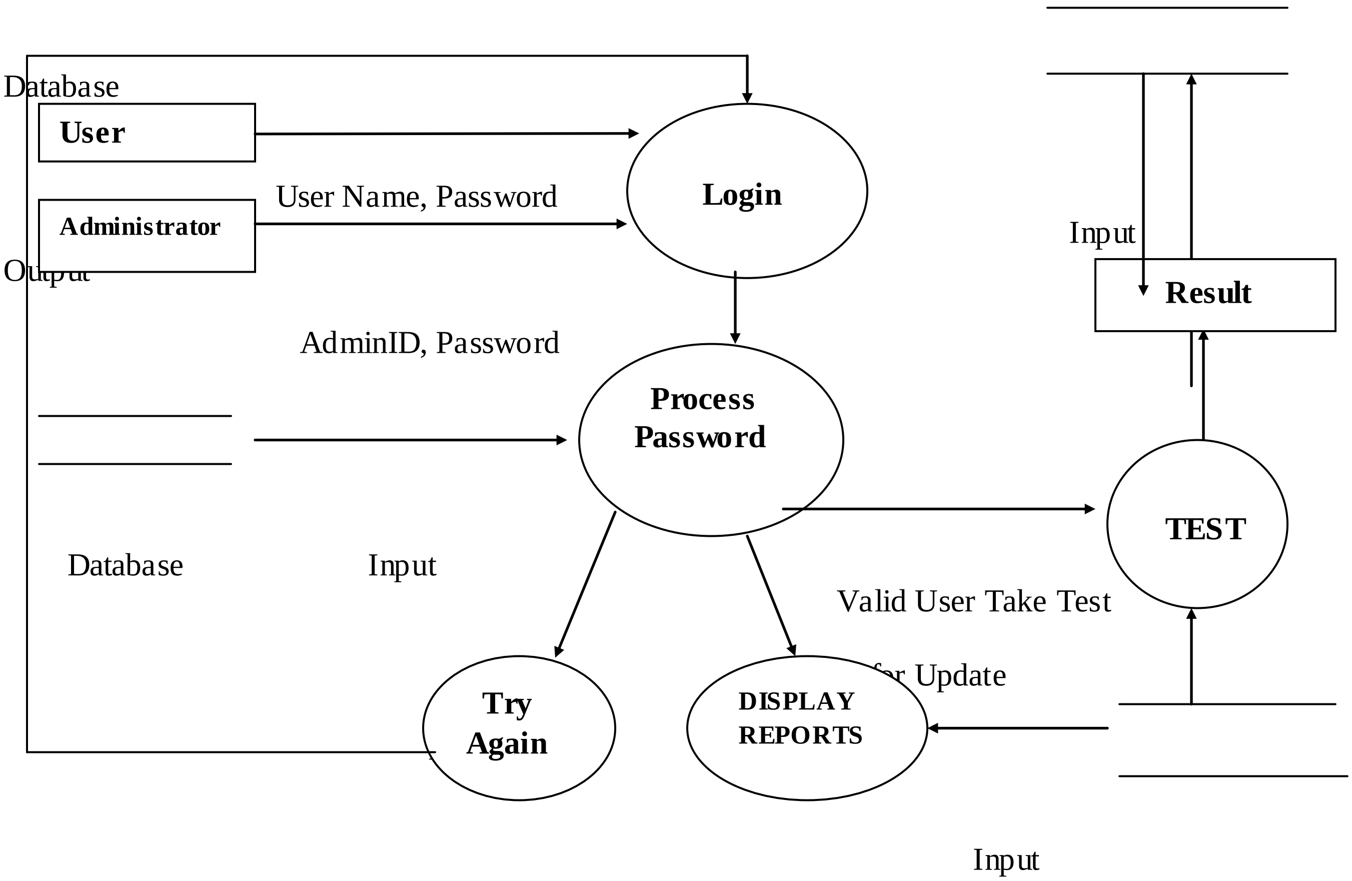
DFD's (Data Flow Diagrams)

Context Level Diagram (0Level)



Note: The names w s not related to any company or college or any p

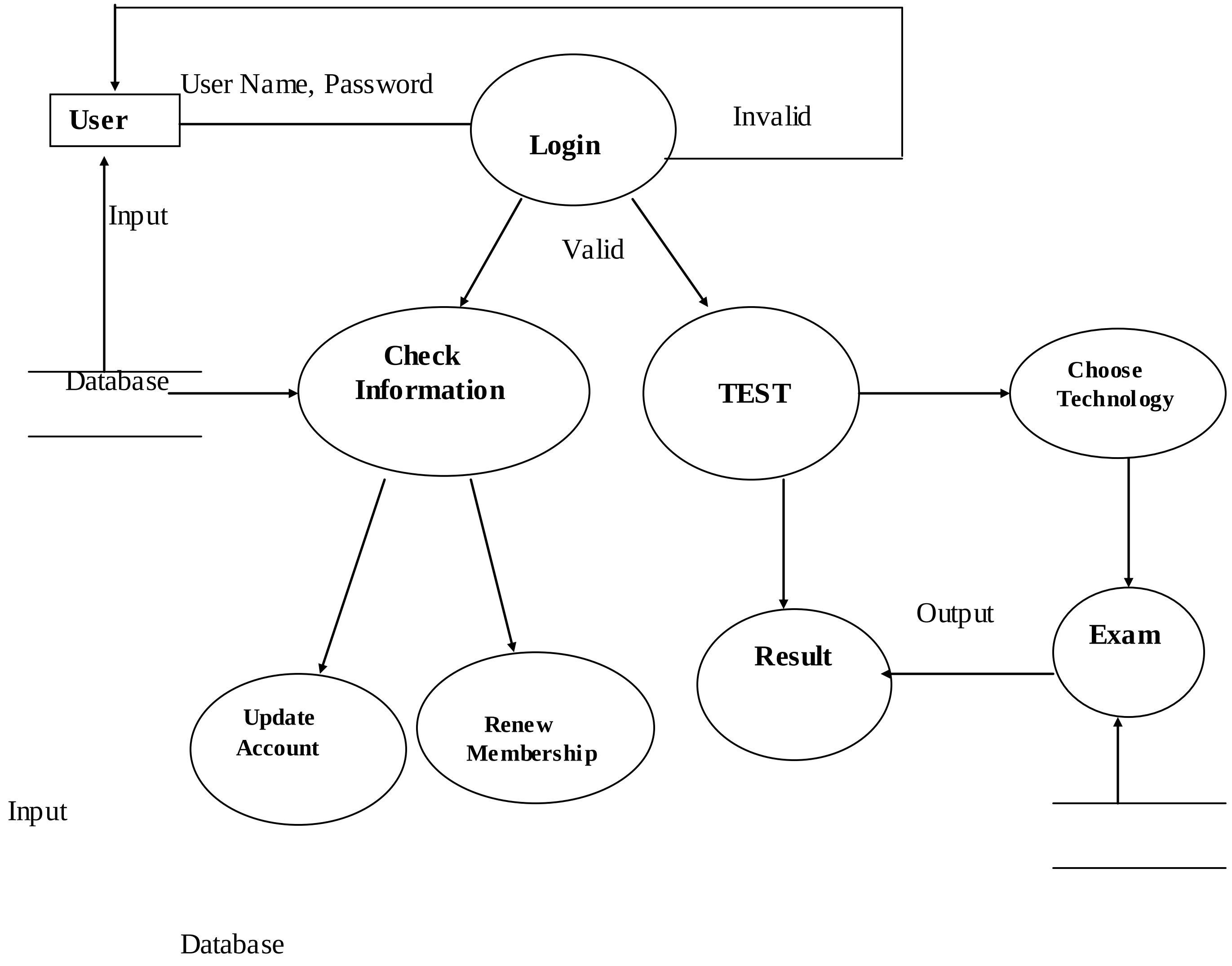
1stLevel Diagram



DATABASE

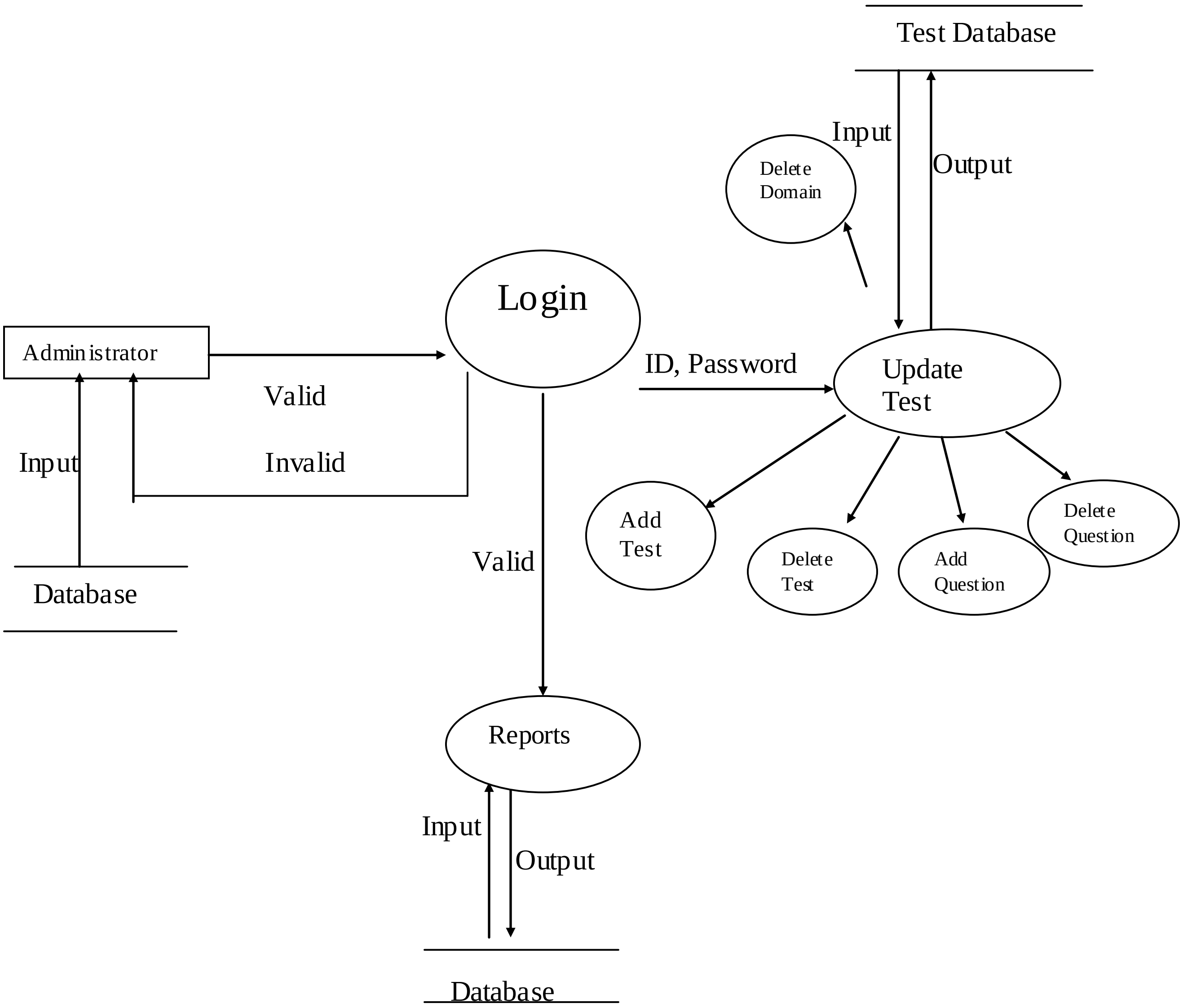
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

2^{end}Level Diagram



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

2^end Level Diagram



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

DATA DICTIONARY

Introduction to data dictionary

Data dictionaries are an integral component of structured analysis, since data flow diagrams by themselves do not fully describe the subject of the investigation. The data flow diagrams provide the additional details about the project/system.

A data dictionary is a catalog- a repository- the elements in a system. These elements center on the data and the way they are structured to meet user requiremen6ts and organization needs. The major elements are data flow, data stores, and processes. The data dictionary stores details and descriptions of these elements.

Entity Relationship Model:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be
  transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the database

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

designer to communicate the design to the end user can use model.

The model can be used as a design plan by the database developer to implement a data model in specific database management software.

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation.

- Entities are represented by labeled rectangles. The label is the name of the entity. Entity

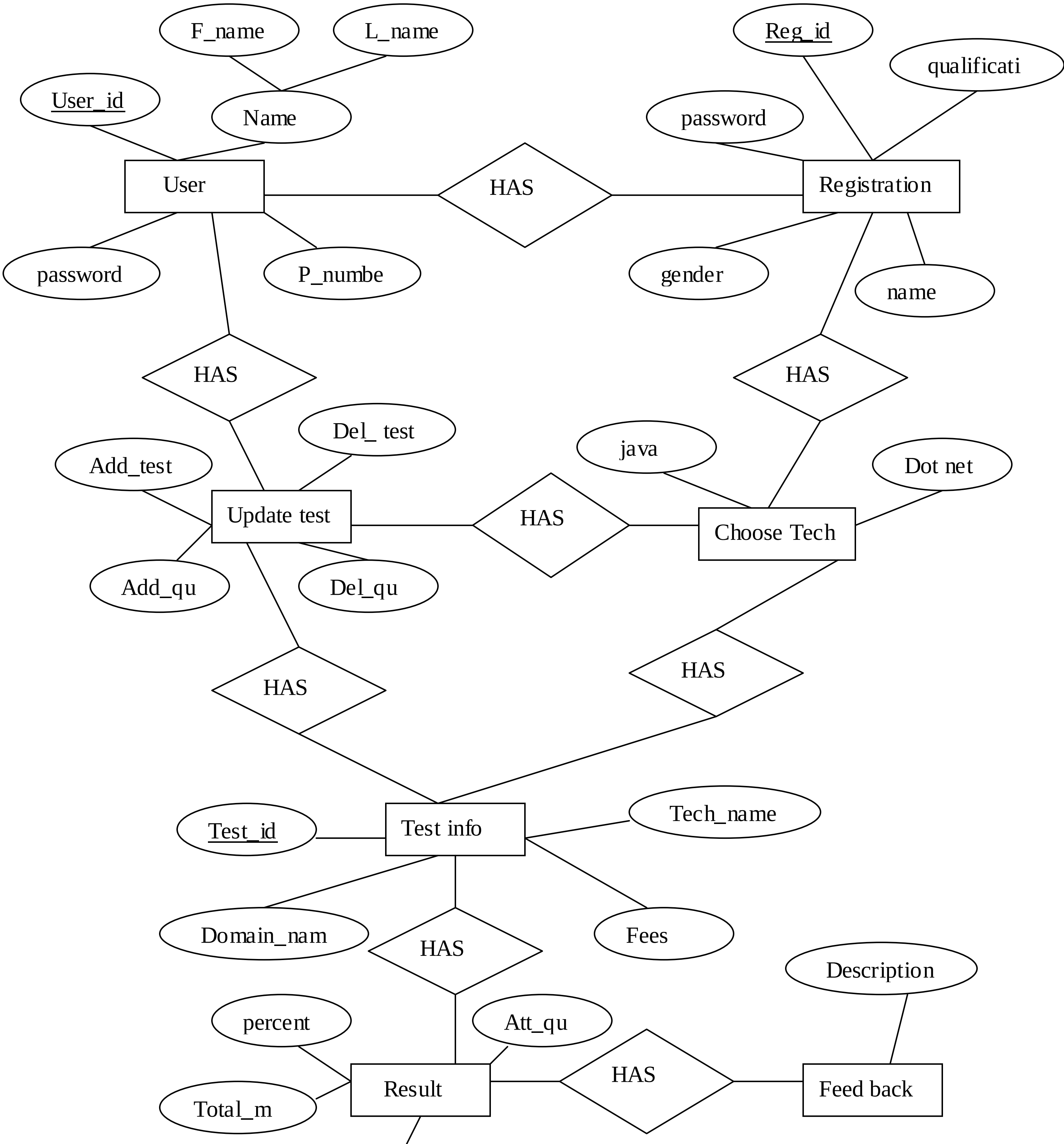
  names should be singular nouns.
- A solid line connecting two entities represents relationships. The name of the relationship

  is written above the line. Relationship names should be verbs.
- Attributes, when included, are listed inside the entity rectangle. Attributes, which are

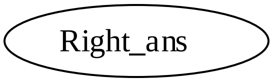
  identifiers, are underlined. Attribute names should be singular nouns.
- A line ending in a crow's foot represents cardinality of many. If the crow's foot is

  omitted, the cardinality is one.

E-R diagram



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.



TESTING

Till now the database design, user interface design and implementation are complete. The system now is tested for its functionality, validity and performance. In order to test the system, a wide variety of tests are conducted to make sure that the system matches the entire identified user requirements and constraints. This chapter focuses on testing the developed systems using different test strategies in order to verify its correctness and user acceptance.

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

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

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. And it needs to be done in almost every phase of product development life cycle not just before a product is handed to a customer.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

The following are some attributes of a good test:

- A good test has a high probability of finding an error. To achieve this goal the tester must
  understand the software and attempt to develop a mental picture of how the software may
  fail. Ideally the classes of failure are probed.
- A good test is not redundant: testing time and resources are limited. There is no point in
  conducting the test that has the same purpose as another test. Every test should have a
  different purpose.
- A good test should be best of breed. In a group of tests that have a similar intent time and
  resource limitations may militate for the execution of only a subset of these tests. In such
  cases the tester that has the highest likelihood of uncovering a whole class of errors should
  be used.
- A good test should be neither too simple nor too complex: although it is sometimes
  possible to combine a series of tests into one test case, the possible side effects associated
  with this approach may mask errors. In general each test should be executed separately.

TYPES OF TESTING

1. Unit Testing

Unit testing focuses verification effort on the smallest unit software design- the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in algorithmic execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths (bases paths) through the control structure are exercised to ensure that all elements in a module have been executed at least once. And finally all error-handling paths are tested.

Application interface of our system was unit tested at all levels of implementation, right from start of code writing, to integrating the code with other modules. Every module was tested fully to check its syntax and logical correctness. Error handling was implemented into relevant modules so that the code doesn't crash on errors.

2. Integration Testing

Integration testing is a systematic technique for constructing the program structures, while conducting test to uncover errors associated with interfacing, the objective is to take unit tested modules and build a program structure that has been dictated by design.

User interface of i-Admit was developed in modules. All of them were joined together to make the complete running application. While integrating these modules, integration testing was performed on them to verify that they meet all interfacing requirements and that they pass relevant information among themselves. In the end the complete program structure was tested to ensure interoperability of all the modules.

3. Validation Testing

At the culmination of integration testing software is completely assembled as a package: interfacing errors have been uncovered and corrected and a final series of software tests – Validation Testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of Black Box tests that demonstrate conformity with requirements.

4. Alpha Testing

It is virtually impossible for a software developer to foresee how the customer will really use a program. When custom software is built for one customer a series of acceptance tests are conducted to enable the customer to validate all requirements.A customer conducts the alpha test at the developer site. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problem. Alpha tests are conducted in a controlled environment. Alpha tests were performed at our development site with the help of our friends, who were called and asked to run the program in the manner they like, without our guidance and errors and usage problems were noted and code was updated to remove all of them.

5. Beta Testing

The Beta test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing the developer is generally not present. Therefore the beta test is a live application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta test the software developer makes modification and then prepares for the release of software product to the entire customer base.Beta testing of our system is not performed as the product is not yet fully developed and has not been installed at the user site as it still is in the development phase. Beta testing will be performed when the software is deployed at the user's site.

6. System Testing

Ultimately software is incorporated with other system elements (new hardware, information) and a series of system, integration and validation tests are conducted. It is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose all work to verify that all system elements have been properly integrated and perform allocated functions.System testing of this system was performed at the development lab of this system by integrating the functional systems to imitate the actual work environment.

.

7. Security Testing

Security Testing attempts to verify protection mechanism built into a system will in fact protect it from improper penetration. Security is provided for each user by giving them login name and password. Security testing was done, as any other anonymous user can't log in with a user password if the user is already logged in.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Future Scope:

i) In future, this site will be interacting with a job site being developed in .NET technology through web services.

ii) The on line exam in today's life is very important ,user give the test at any time and

any where no need for giving the exam by specific date so user will be take benefit by

on line exam.

iii) In coming days all students are certified and educated by this site and there certification will be Recognized in the word.

iv) The most reputed company is also registering its sites so they can recruit on behalf of

the test marks.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

CHAPTER 1

INTRODUCTION

OBJECTIVES

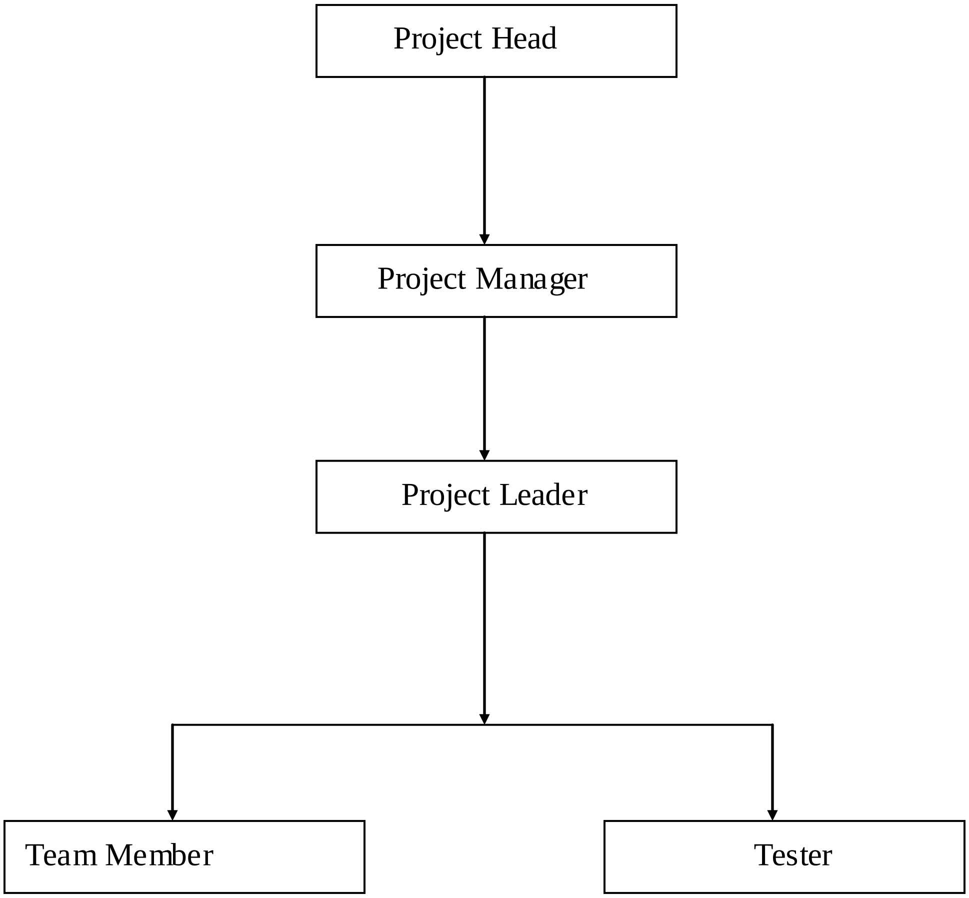
1 The main Objective of my project (Online Examination) is to replace tiring paper work involved in the manual Examination management system. The project aims at creating such software, which can help examination management Staff to maintain and search the any types of information in the form of different records on a Personal Computer. It is an objective based test which can be given by any graduate student staying any where in the country? As it is online examination so it is operated through net.

SOFTWARE DEVELOPMENT:

- Microsoft
- Sun Microsystems
- CISCO
- Novel
- Oracle

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Organization Structure



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

1.3 OBJECTIVE AND SCOPE OF THE PROJECT

On Line Examination System is a website in which the many Students simultaneously appeared in the exam. The exam can be of different technology and the question paper will be randomly generated. The evaluation of all the papers is done online by the software. This Project is to be developed for a client High technologies Solutions Noida

Objective:

Online Examination System, System will hold ONLINE tests in different technology to candidates and evaluate their scores in various tests and store the results of registered candidates after that provided certified professional certificate. Further, He or She who is searching Jobs in MNC This score which have got in this Test, Help for getting job and The all Companies are given the values of this scores So the user will be ahead to all the competitors and user will be get benefit in the company's Interview .

.

Future Scope:

iv) In future, this site will be interacting with a job site being developed in .NET technology through web services.

v) The on line exam in today's life is very important ,user give the test at any time and any where no need for giving the exam by specific date so user will be take benefit by on line exam.

vi) In coming days all students are certified and educated by this site and there certification will be Recognized in the word.

iv) The most reputed company is also registering its sites so they can recruit on behalf of the test marks.

1.4 APPROACH

Using system approach to develop information system which is known as system development life cycle . It is a step by step process SDLC refers to a methodology for developing systems. It provides a consistent framework of tasks and needed to develop systems. The SDLC methodology may be condensed to include only those activities appropriate for a particular project, whether the system is automated or manual, whether it is a new system, or an enhancement to existing systems.

The SDLC methodology tracks a project from an idea developed by the user, through a feasibility study, systems analysis and design, programming, pilot testing, implementation, and post-implementation analysis. Documentation developed during the project development is used in the future when the system is reassessed for its continuation, modification, or deletion.

System Development Life Cycle.

The Systems Development Life Cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved including the waterfall model (the original SDLC method), rapid application development (RAD), joint application development (JAD), the fountain model and the spiral model. Mostly, several models are combined into some sort of hybrid methodology. Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with the development process. Some methods work better for specific types of projects, but in the final analysis, the most important factor for the success of a project may be how closely particular plan was followed.

The software development life cycle (SDLC) is the entire process of formal, logical steps taken to develop a software product. The phases of SDLC can vary somewhat but generally include the following:

Phase 1 - Study and Analysis Phase

In this phase, the data, facts and figures will be gathered by checking various documents related to system. It also covers interviews of the persons who belong to the system. The problems in the current system will be identified and necessary improvements will be recommended. Thus a new system, as a solution will be proposed. The emphasis is on the type of information, the new system should provide, the required frequency and accuracy of results and other considerations. The activities of analysis phase will be least structured and more people oriented.

Phase 2 - Design Phase

The design of a system produces the details that state how a system will meet the requirements identified during system analysis. In this phase, the detailed specifications for the new system will be formulated. They will describe its features: the outputs, inputs, files and databases, and procedures – all in a manner that meets project requirements.

Following the analysis of the existing system, the next step is to develop the Information System Architecture and Enterprise Data Model (EDM) that will describe effectively the existing system. The conceptual model will lead to the development of an Enhanced Entity Relationship Diagram. This will be followed by the development of relational model that describes the tables (or relations) that can be suitably implemented in any modern relational DBMS software like SQL

Phase 3 - Coding and Implementation Phase

The activity following logical design, which produces program software, files and a working system, will be carried out. This process will be initiated by first identifying the key factors responsible for the layout of the software. Then many suggestions received at a previous stage are analyzed and categorized according to their nature. Then the solutions of these problems will be designed and developed.

The physical model will cover all the physical file organization issues leading to the actual relations that will be implemented on to the system. The Query Implementation will demonstrate the various queries that will be executed on the relations created during design phase. This will be followed by the development of user interfaces and their connectivity with the database.

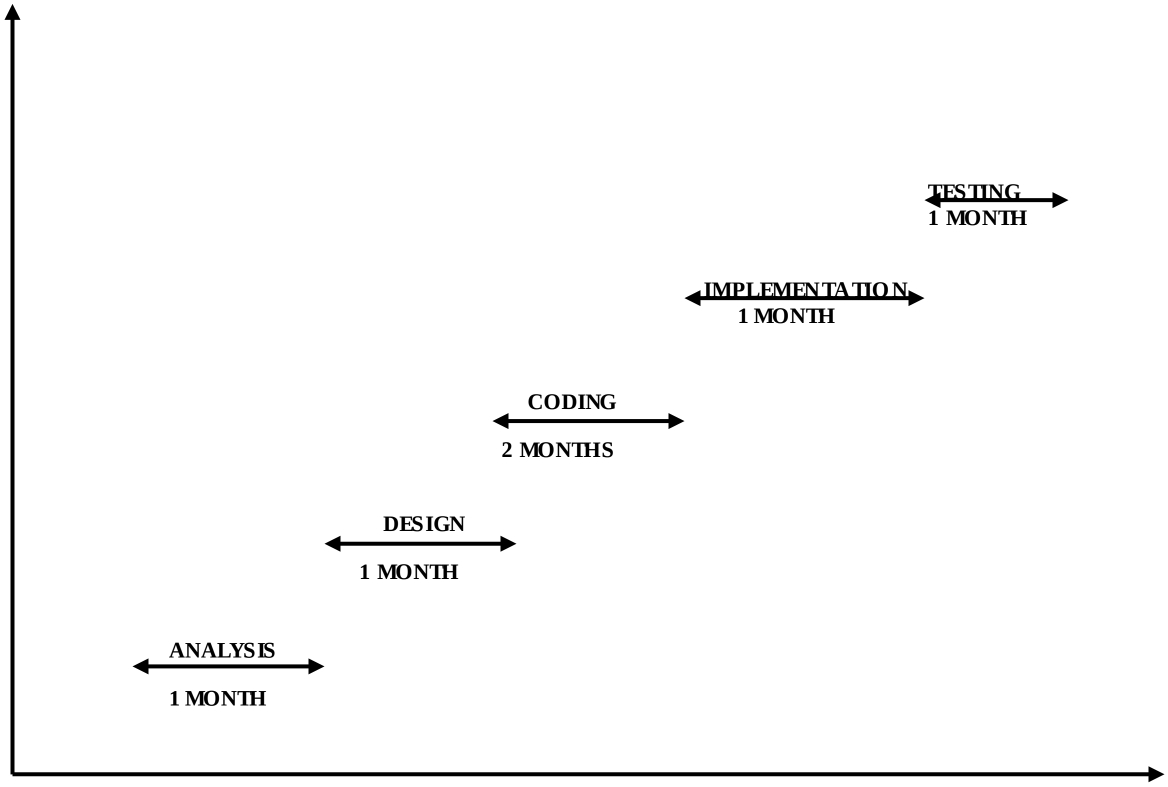
Phase 4 – Testing

Till now the database design, user interface design and implementation are complete. The system now is tested for its functionality, validity and performance. In order to test the system, a wide variety of tests are conducted to make sure that the system matches the entire identified user requirements and constraints. This chapter focuses on testing the developed systems using different test strategies in order to verify its correctness and user acceptance.

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

1.5 TIME SCHEDULING

Automated tools, specialized notations and modern techniques are often used to develop software requirement specifications, architectural and detailed designs and the source code. Management tools such as PERT, Gantt charts, work breakdown structures and personnel staffing charts may be used to track and control progress.



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

CHAPTER 2

SYSTEM ANALYSIS

Systems Development can generally be thought of as having two major components: Systems Analysis and Systems Design. Systems Design is the process of planning a new business system or one to replace or complement an existing system. But before this planning can be done, we must thoroughly understand the old system and determine how computers can best be used (if at all) to make its operation more effective. Systems analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system.

2.1 Current System Summary

FEASIBILITY STUDY:

It means to carry out the detailed study of the existing system, find out the problem related to technical, operational staff, and economical field like cost by concerning the user of the study proposal of the proposed system (i.e. creditor ledger) is accepted by the management it will lead to the investigation of the existing system or problem area. Having recognized the problem, the next step to carry overall analysis of the system requirement in terms of it's:

- The input data
- Type of processing needed
- The output

Three phases of the feasibility study is used

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Technical feasibility:

It is concerned with the available hardware and the software resources whether they meet the given requirement of the analyzed system or not which include latest machinery and the technique required handling the system. It may also invoke the study of the new alternative to solve the given problem.

Behavioral Feasibility:

People are inherently resistant to change, and computers have known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover, transfers, restraining, and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell and train the staff on new ways of conducting business.

Economical feasibility-

It deals with the study of the cost benefit analysis. All the cost of the new system compared with the benefits, which can be obtained for management approval. The benefit may be quantities in nature Current System Summary. The genuine consideration of the system being developed is the approach we follow to look the system in the way it is useful for us.

2.1.1 Background

Provide background information concerning the uses and purposes of the current system. Refer to interfacing systems when needed to enhance the general description.

2.1.2 System Objectives and Current Functionality

Modules:

OLSE Project includes the following modules:

6) User

7) Registration

8) Administrator

9) Test

10) Database Design

5) **User & Registration**: User module is related with the users of the OLSE system.

- There are three types of users of the system, users who wants to take test, admin of the system, and guest users for demo.
- A guest user can only take sample tests, and access certain areas of site.
- A guest user can register oneself with the site and avail the facilities provided by the site.
- A user can register oneself under different membership types. Membership types available are:
- Platinum
- Golden
- Silver
- Administrator can log into the site. He/she will have complete control and access to Each part of the site
- There will be a payment plan for the users of the site (quarterly, annually, Semi – annually).

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

- The payment will be based on membership types.
- If a user applies for certification, the payments will differ.
- Payment mode for any type of payment will be credit/debit card.
- Certificates will be delivered to the user after payment has been made.
- User can ask queries.
- User can fill the feed back form
- A registered user will login into the system.
- A session of the registered user will be maintained
- After logging in to the system, a user can view his/her profile.
- A registered can view all the tests.
- A registered user can apply for a test.
- A registered user can change his/her personal details.
- A registered user can view public transcripts.
- A registered user can view his/her private transcripts.
- A registered user can view scores in a particular test.
- A registered user can view overall performance.
- An administrator will be displayed admin related tasks.

6) Administrator

- Admin module relates to administrative tasks.
- It also deals with admin login.
- The admin can view all the information related to site after logging in.
- He can add new tests.
- He can grant and revoke privileges to certain users of site.
- Admin will enjoy all the privileges related to an admin.

7) Test

- A registered user who wants to take test applies for test.
- He/she can take tests as per his/her wish.
- A registered user can select various subjects and their topics on which he/she wants to

test himself/herself.

- Their scores will be calculated and displayed to them
- He/She can fill the feed back form.
- These results will be stored in the database.

8) **Database Design**

- This module is related with identification of tables that is to be used in the project.
- The project leader with software designers will be responsible for identification of Tables, their attributes and relationship amongst them.

2.1.3 Current Methods and Procedures

Briefly describe the current methods and procedures being employed to satisfy the existing information requirements. Provide a graphic representation that depicts the existing data flow through the functional system from data acquisition through its processing and eventual output. The graphic may be complimented by a narrative explanation of the sequence in which the user performs the operational functions.

2.1.4 Input and Output

- A guest user can only take sample tests, and access certain areas of site.
- A guest user can register oneself with the site and avail the facilities provided by the site.
- A registered user will login into the system.
- A registered can view all the tests.
- A registered user can apply for a test.
- A registered user can change his/her personal details.
- A registered user can view his/her private transcripts.
- A registered user can view scores in a particular test.
- A registered user can view overall performance.

- An administrator will be displayed admin related tasks
- . Their scores will be calculated and displayed to them
- He/She can fill the feed back form.

These results will be stored in the database

2.1.5 Deficiencies

2.2 PROPOSED METHODS AND PROCEDURES

* The proposed method is clicking the links which direct a page to another page. Methods that used in project are link reference bind with image link, button link and static link.

* Another procedure that used project is search method written by coder and as well we make a search method by using Google search engine.

2.2.1 Summary of Improvements

The major improvement provided by the system is that the accessing the software is easy and any layman can also easily use the system.

System analysis is conducted with the following objectives in mind:

1. Identify the user’s need.
2. Evaluate the system concept for feasibility.
3. Perform economic and technical analysis.
4. Allocate functions to hardware, software, people, database and other system elements.
5. Establish cost and schedule constraints.

Explicitly identify the requirements to be satisfied by the proposed system

2.2.1.1 Functional Improvements

In this section, we discuss the improvements, which are held after developing the system. And we consume how we will achieve the functional improvements. There are various factors due to which we can show the improvements, some of the improvements are as follows-

- The first improvements that comes, which is it is less time consuming. Due to new system, there is less paper work.

- The most striking improvement is that it increases the accuracy and makes data more validate

2.2.1.2 Improvement to Existing Capability:-

There are various improvements that are intended to the existing Capability. Since at beginning the work that are done, is totally manually but when the system is computerized then it becomes a time saving technique and the all the records are maintained in the very less time than the manual system and there may be taken a short idea in few minute of all work

This System is very beneficial to the user because of the following reasons.

1. This helps to get rid of from pen and paper work.
2. This helps to decrease redundancy, which is in the file system.
3. Institute's efficiency and effectiveness will increase because this will give the quick and accurate output of the user's choice.
4. Accuracy will be increased because institute can use the efficient information provided by the system at right time and can take effective decision for the betterment for future of student

Time can be saved which can be utilized in some other needs

2.2.1.3 Timeliness:-

The proposed system is real time processing system. It updates the database when any transaction is submitted to the system. Due to this the master file is always up-to-date, so no problem to the user. In the existing system the data are processed in the batch so the master file is not up-to-date. It creates a problem to the user sometimes. The system is automated so it gave fast response to the user. Explicitly identify the requirements to be satisfied by the proposed system.

2.2.2 Summary of Impacts

Summarize the anticipated impacts and associated costs (detailed in the following subsections) of the proposed system on the existing organizational and operational environments of the user, as well as to the user during the development of the system.

2.2.2.1 User Organizational Impacts

No special training is required to teach the user about accessing the software and thus it saves the time and money required for training of the personnel.

2.2.2.2 User Operational Impacts

It is easy to work on the software. And forms designed are user friendly that transaction can easily be done through the software.

2.2.2.3 User Developmental Impacts:-

To operate this system user just need to know the simple knowledge of Internet and Basics of computer operation

2.3 DETAILED CHARACTERISTICS

2.3.1 Specific Performance Requirements

Describe the specific performance requirements for the system as a whole and for major functions or subsystems within it. Delineate the requirements on which the system design will be based. Include a quantitative presentation of requirements, such as the number of events that must be processed, maximum allowable time from query to receipt of requested information, and flexibility required to accommodate changing user requirements.

2.3.1.1 Accuracy and Validity:

Describe the accuracy requirements to be imposed on the system. Accuracy requirements of mathematical calculations and data must be considered.

2.3.1.2 Timing:

Traditional system was acquiring a lot of time but in opposite the existing system is producing a faster response in every area of the system for which it has been developed.

1. Response time is faster to queries and updates
2. Sequential relationship of functions

2.3.1.3 Capacity Limits:-

Specify the maximum numbers of transactions, storage requirements, concurrent users, or any other quantifiable information about capacity requirements placed on the system. Identify changes to capacity limits resulting from varying modes of operation. Include peak load limits and issues

2.3.1.4 Reason for Choosing Asp.Net as a Front End

Introduction the .NET Platform:

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.

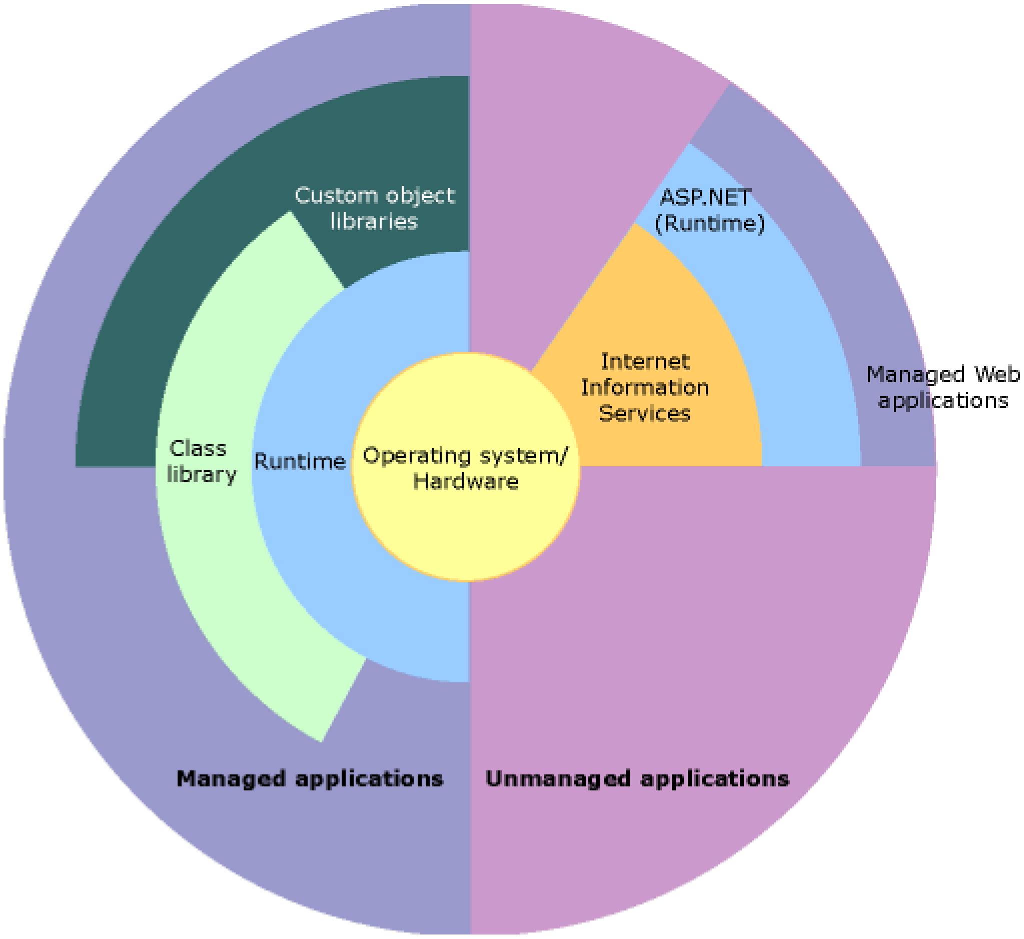
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates 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. The runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

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

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

.NET Framework in context



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Introduction to ASP.NET

ASP.NET is more than the next version of Active Server Pages (ASP); it is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. While ASP.NET is largely syntax compatible with ASP, it also provides a new programming model and infrastructure for more secure, scalable, and stable applications. You can feel free to augment your existing ASP applications by incrementally adding ASP.NET functionality to them.

ASP.NET is a compiled, .NET-based environment; you can author applications in any .NET compatible language, including Visual Basic .NET, C#, and JScript .NET. Additionally, the entire .NET Framework is available to any ASP.NET application. Developers can easily access the benefits of these technologies, which include the managed common language runtime environment, type safety, inheritance, and so on.

ASP.NET has been designed to work seamlessly with WYSIWYG HTML editors and other programming tools, including Microsoft Visual Studio .NET. Not only does this make Web development easier, but it also provides all the benefits that these tools have to offer, including a GUI that developers can use to drop server controls onto a Web page and fully integrated debugging support.

Developers can choose from the following two features when creating an ASP.NET application, Web Forms and Web services, or combine these in any way they see fit. Each is supported by the same infrastructure that allows you to use authentication schemes, cache frequently used data, or customize your application's configuration, to name only a few possibilities.

Web Forms allows you to build powerful forms-based Web pages. When building these pages, you can use ASP.NET server controls to create common UI elements, and program them for common tasks. These controls allow you to rapidly build a Web Form out of reusable built-in or custom components, simplifying the code of a page. For more information, see Web Forms Pages. For information on how to develop ASP.NET server controls.

2.3.1.5 Reasons for Choosing SQL 2005 as Back End

A database system is basically a computer based record keeping system. We have chosen sql server as Database System because it is easily available as it comes part of sql suite.
The Sql Server 2005 relational database engine supports the feature required to support demanding data processing environments .The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the databases

2.3.2 Transaction Requirements

Basically the Transactional Requirements of a Online skill examination systems
are requirement student login name and password.

2.3.3 External interface requirements

2.3.3.1Hardware interfaces

MINIMUM HARDWARE

It's a web – based project, so a robust hardware configuration is required. The hardware requirements are:

i) Processor: P IV 2.4 GHz and above.

ii) Motherboard: Intel 845 and above

Note: The names we used in this it is a dummy name and its not related to any
company or college or any person.

iii) RAM:	512MB
iv) Hard Disk:	3 GB for .NET and 2 GB for Microsoft SQLServer2005.
v) Network Card:	Standard Ethernet card for networking.
vi) I/O Devices:	Keyboard, mouse and Color monitor
vii) Wires:	Twisted pair for networking.

2.3.3.2 MINIMUM SOFTWARE

Following software are required for developing OLSE application:

i) Operating System:	Windows NT, 2000, XP Prof Etc.
ii) Platform:	ASP.NET, C#.
iii) Environment:	VisualStudio.NET 2.0
iv) Tools:	IIS
v) RDBMS:	SqlServer2005

2.3.3.3 Communications interface

Connection of database to ASP.NET forms is done through ADO.NET connection.

Describe by individual function the major functional processing steps. This may be done using graphic representation (e.g., flowchart and DFD) or descriptive use cases. Provide enough detail to support development of design specifications.

The production flowchart is a visual representation of the content of the sequence of the content of your product. It show what comes first, second, third etc. as well as your audience will do, if anything, and what will happen when they have done it. A completed flowchart organizes your topics; strategies, treatments & options into a plan from which you can work out the details of what each screen, page, or shot will look like.

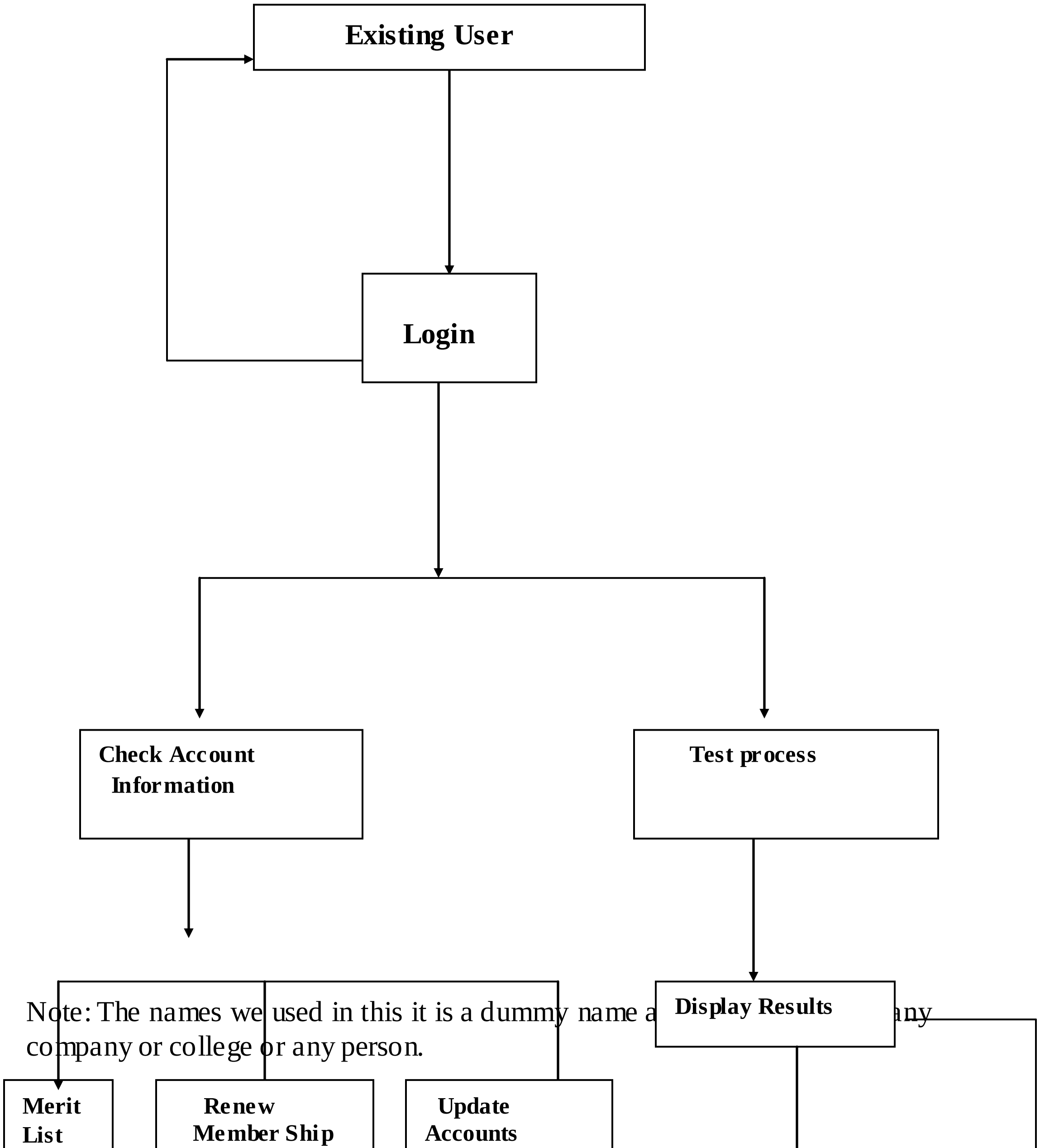
Quality improvement tool:

Flowchart used specifically for process.

1. A flowchart is defined as a pictorial representation describing a process being studied or even used to plan stages of a project. Flowchart tends to provide people with a common language or a reference point when dealing with a project or a process.
2. Four particular types of flowcharts have proven useful when dealing with a process analysis: top-down flowchart, detailed flowchart, work
3. Flowchart and a development chart. Each of the different types of flow charts tends to provide a different aspect to a process or a task. Flowcharts provide an excellent form of documentation for a process, and quite often are useful when examining now various steps in a process work together.
4. When dealing with a process flowchart, two separate stages of the process should be considered: the finished product and the making of the product. In order to analyze the finished product or how to operate the process, flowcharts tend to use simple and easily recognizable symbols. The basic flowchart symbols are used when analyzing how to operate a process.

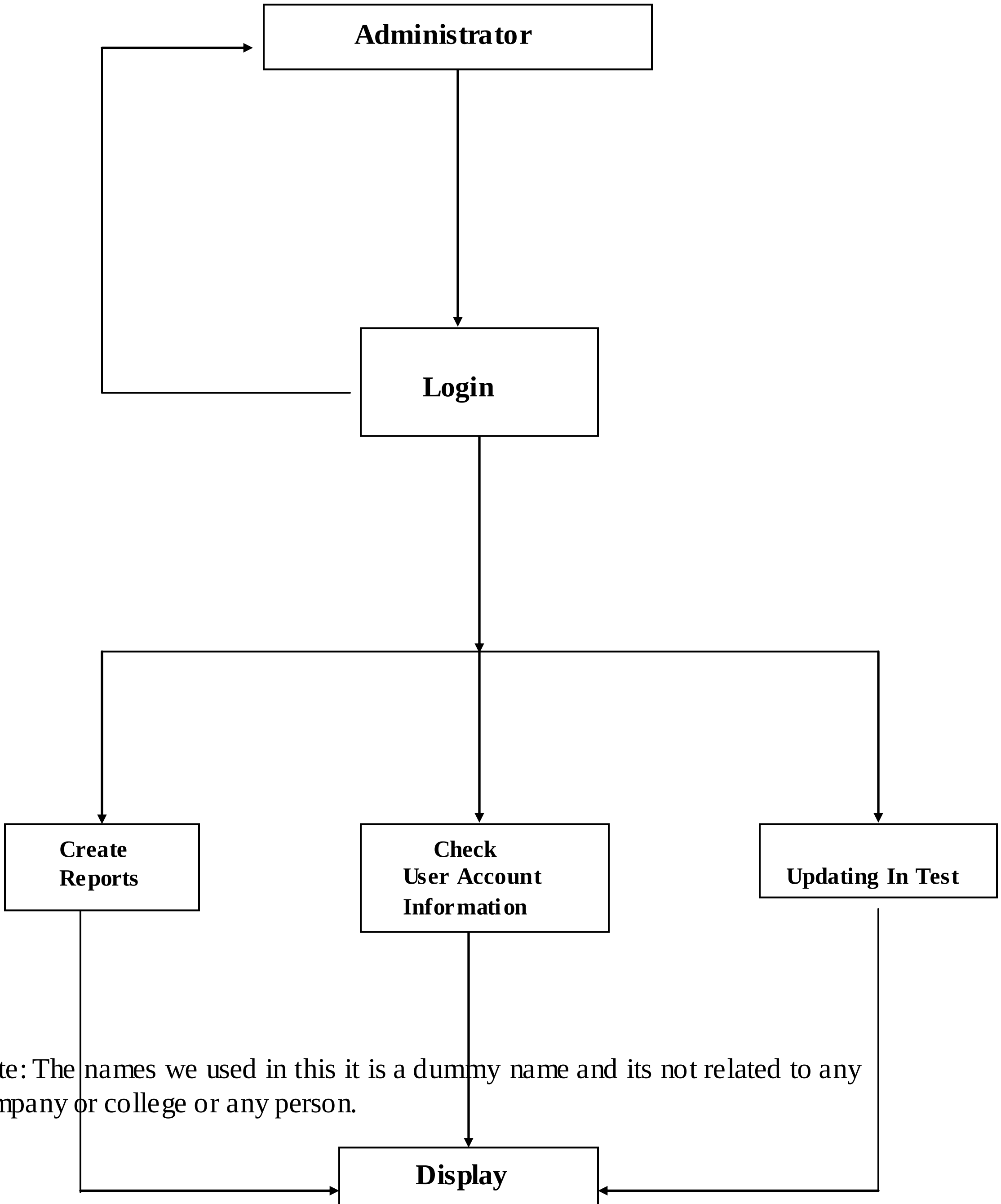
2.4.1 System Flow Chart

Process logic for user module:



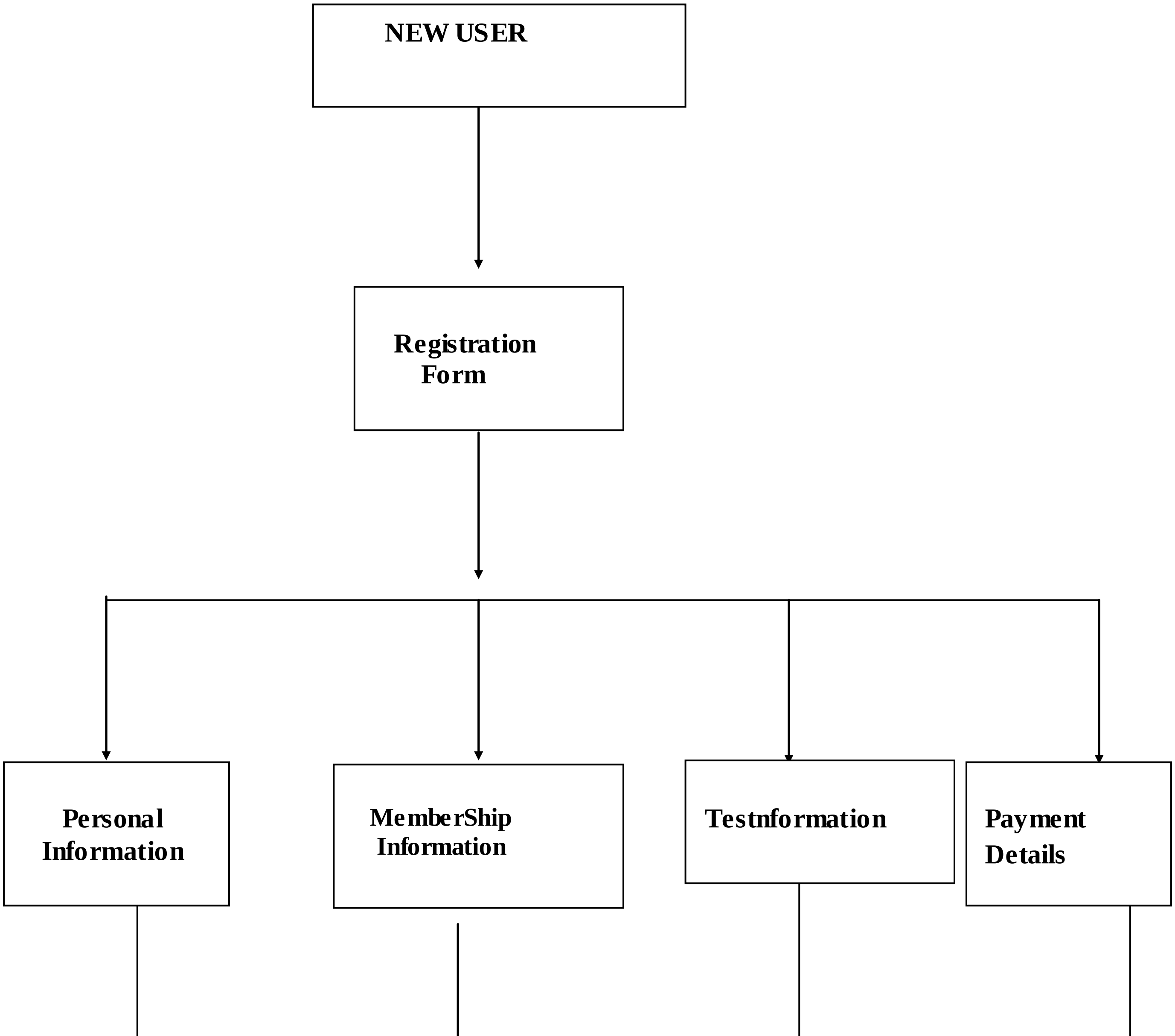
Note: The names we used in this it is a dummy name a any company or college or any person.

Process logic for Administrator module:



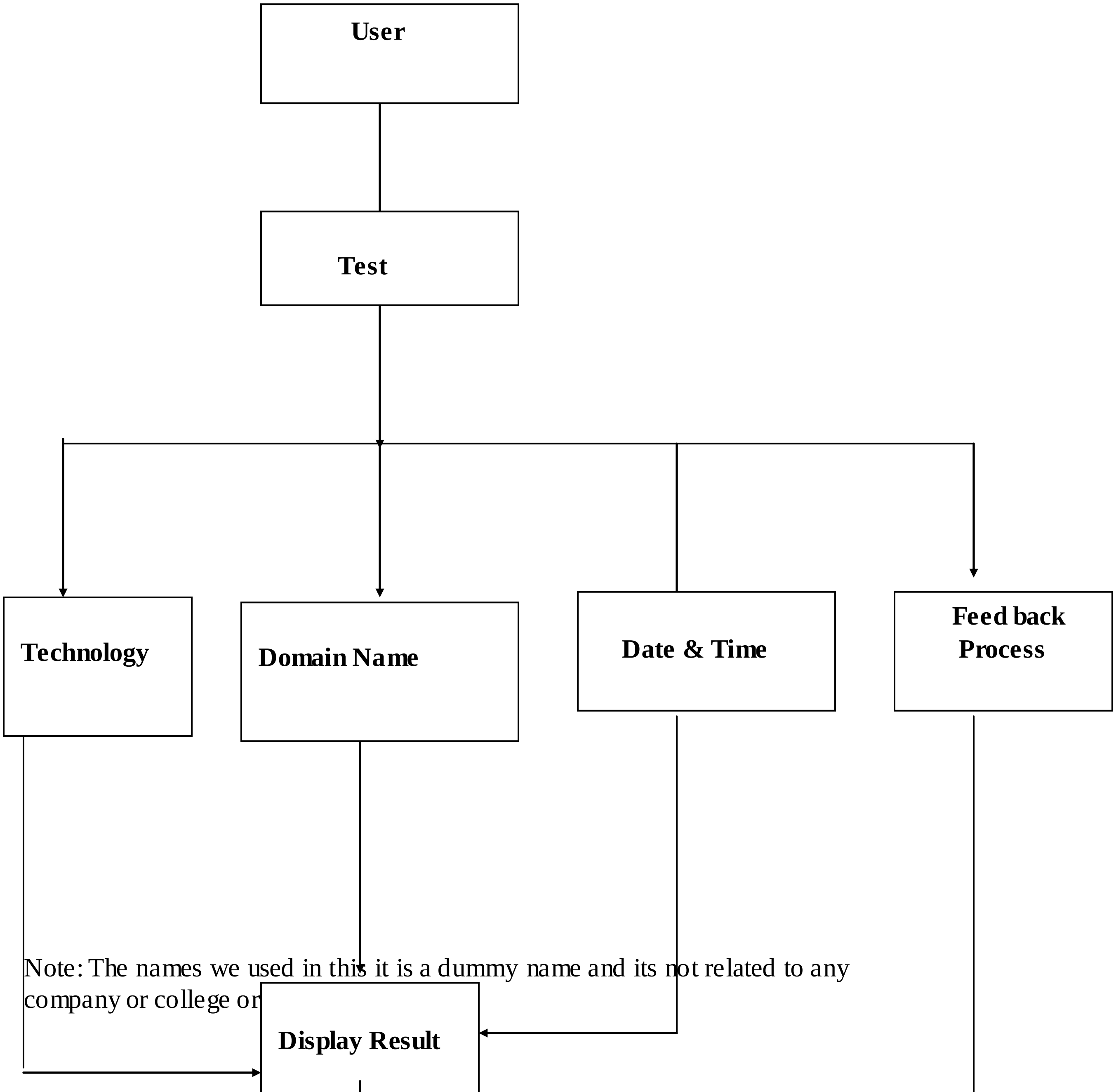
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Process logic for Registration module:



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Process logic for Test module:



Note: The names we used in this it is a dummy name and its not related to any company or college or

2.4.2 Data flow diagram

The Data Flow Diagram (DFD.) was first developed by 'Larry Constantine' as a way of expressing system requirements in a graphical form. D.F.D. represents "what data flow rather than how they are processed".

DFD knows as "bubble chart", has a purpose of clarifying system requirements and identifying major transformation that will become the programs in system designs.

A DFD. consists of a series of bubbles joined by lines**.** The bubble represents data transformation & line represents the data flow in the system.

DFD Symbols

1. A square defines a source or destination of system data.
2. An arrow identifies data flow, data in motion.
3. A circle or bubble represents the process.
4. An open rectangle represents data source.

It does not depend on hardware, software, data structure of file organization.

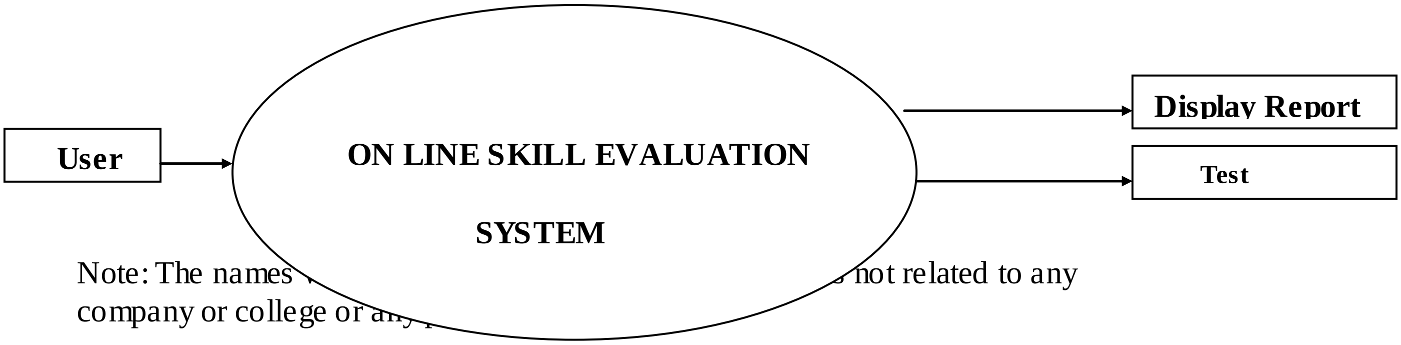
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

CONTEXT DIAGRAM

Diagram plays a very important role in studying the current system .It contain a single process which the system interface to the library or can say it determines the boundaries. It is a top level DFD.

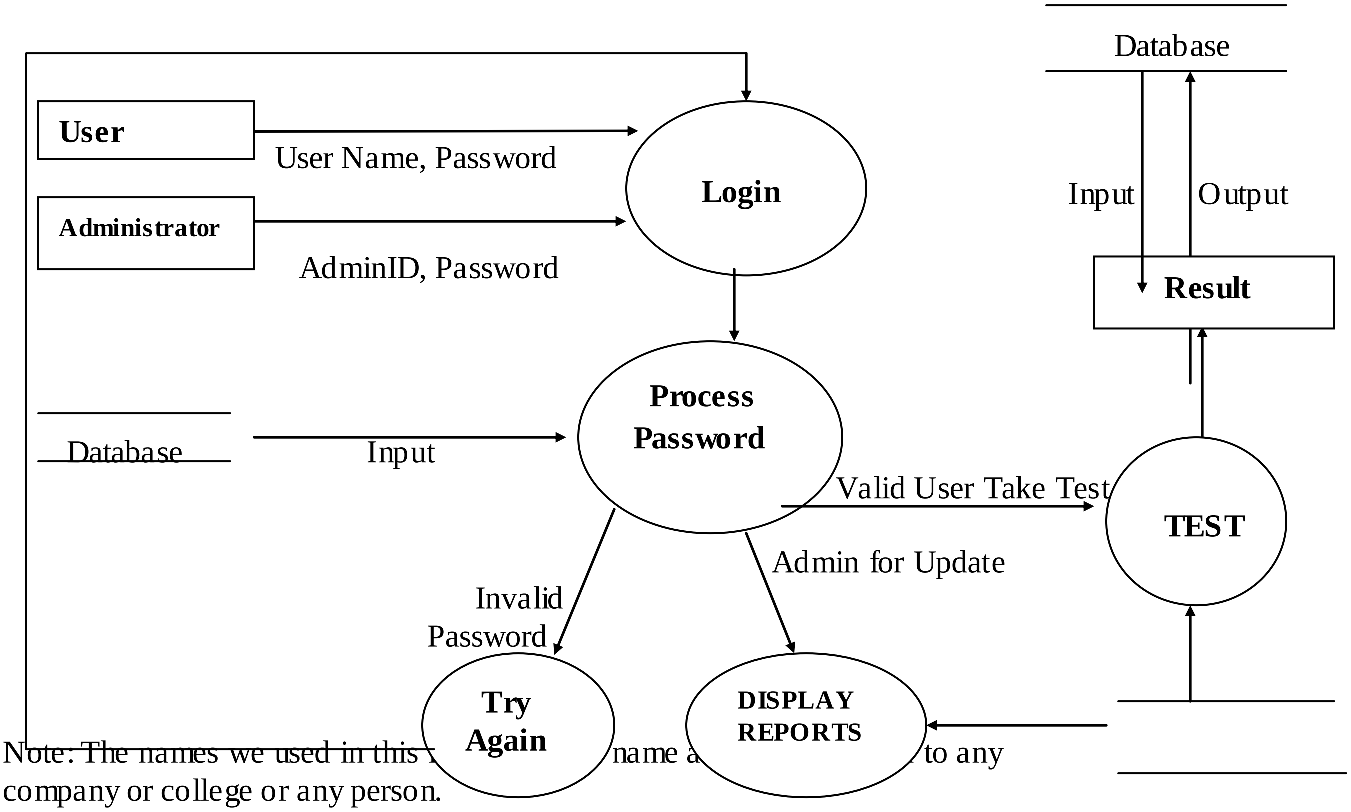
DFD's (Data Flow Diagrams)

Context Level Diagram (0Level)



Note: The names ... not related to any
company or college or any ...

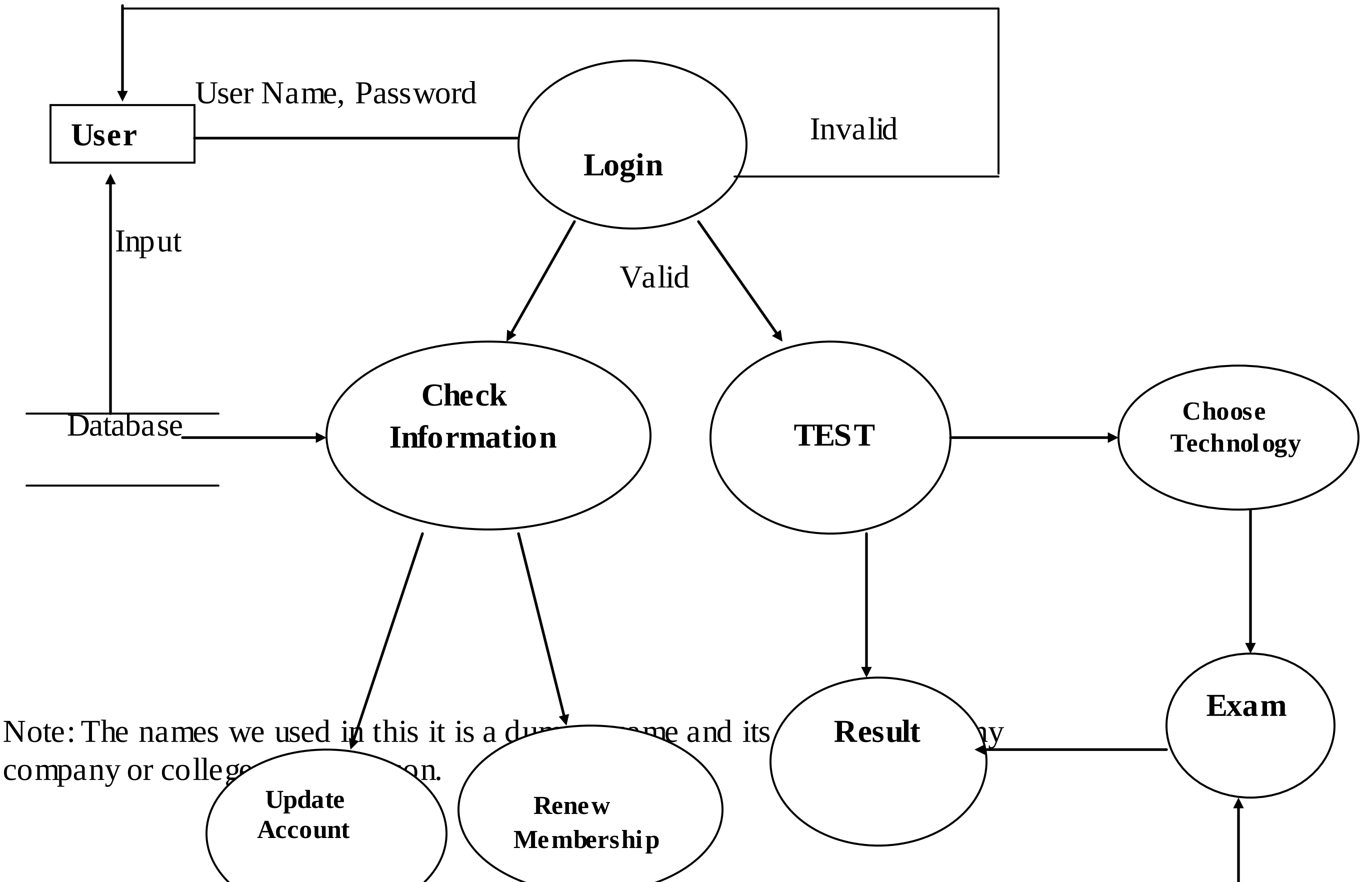
1st Level Diagram



Note: The names we used in this name a to any company or college or any person.

Input DATABASE

2^{end}Level Diagram



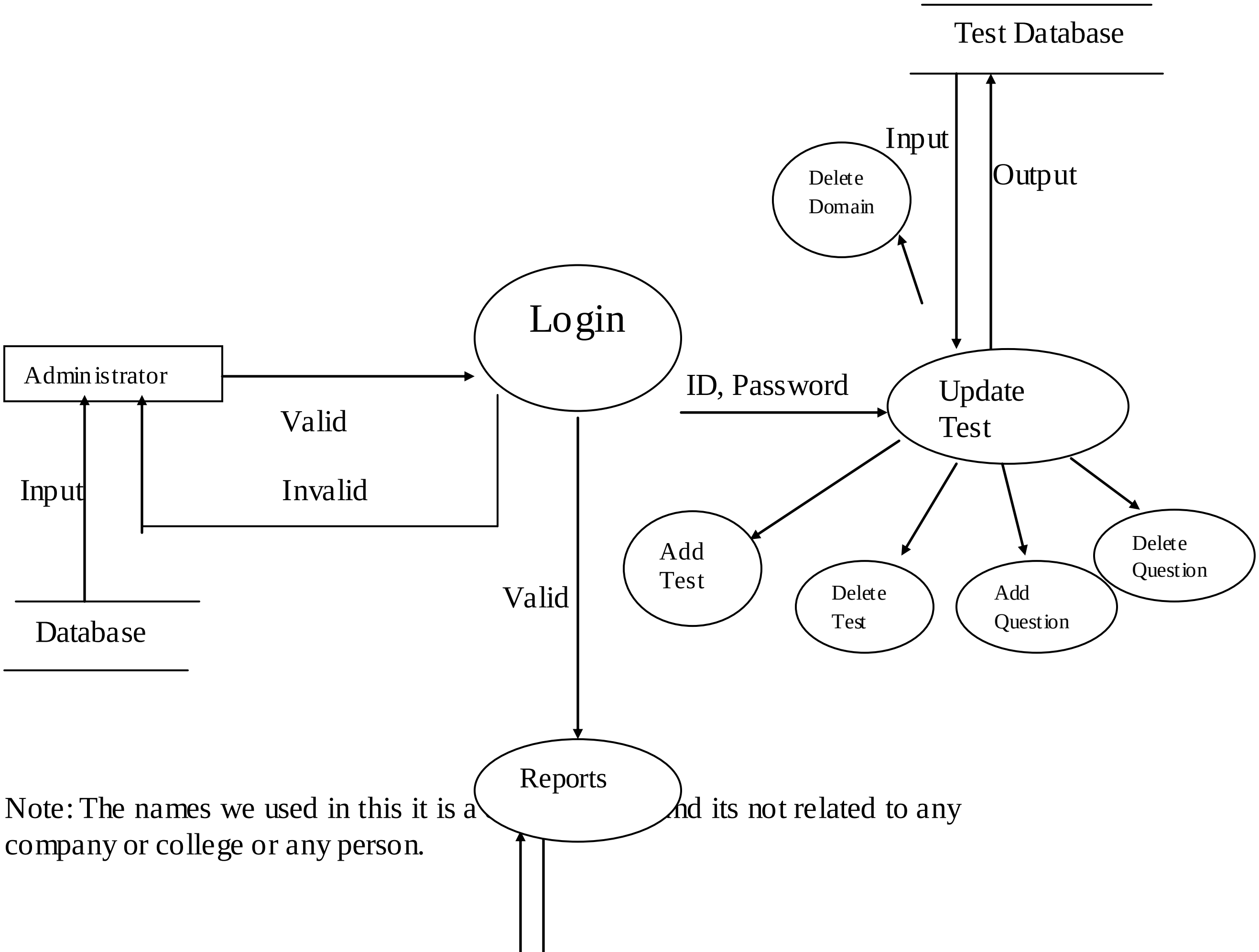
Note: The names we used in this it is a du me and its ny company or colleg on.

Output

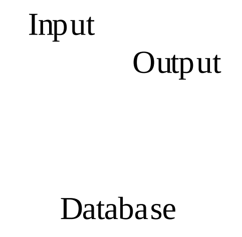
Input

Database

2^{end}Level Diagram



Note: The names we used in this it is a nd its not related to any company or college or any person.



2.4.3 DATA DICTIONARY

Introduction to data dictionary

Data dictionaries are an integral component of structured analysis, since data flow diagrams by themselves do not fully describe the subject of the investigation. The data flow diagrams provide the additional details about the project/system.

A data dictionary is a catalog- a repository- the elements in a system. These elements center on the data and the way they are structured to meet user requiremen6ts and organization needs. The major elements are data flow, data stores, and processes. The data dictionary stores details and descriptions of these elements.

2.5 Input and Output Requirement

Provide a general description of each of the batch and online inputs and outputs. Include information regarding the following:

- Reports and queries to be generated by the system

2.6 Failure Contingencies

Note: The names we used in this it is a dummy name and its not related to any
company or college or any person.

The alternative courses of action that are to be taken to satisfy the information requirements if the proposed system fails. Include as appropriate:

- Backup: Identify backup requirements for ensuring the continued achievement of system functions.
- Fallback: Identify fallback techniques for ensuring the continued satisfaction of the specific requirements of the system.

2.7 Summary

In this chapter, the existing system has been thoroughly studied. The functioning of the Institute is understood and documented in detail. The transactional and input/output requirements are written. The various problem areas are discussed and their solutions are presented.

CHAPTER 3
Conceptual and Logical Database Design

3. Conceptual Design

As of now, the organization has been described with the development of the information system architecture and the Enterprise Data Model (EDM). This chapter carries out conceptual design, which will take over from the external schema in the database development procedure. This chapter will finally lead to the development of the Entity Relationship (ER) Model by defining the entities and their relations in detail followed by describing the structural constraints.

A Conceptual schema is a detailed specification of the overall structure of organizational data, while being independent of any database management technology. A conceptual schema defines the whole database without reference to how data are stored in a computer's secondary memory. A Conceptual Schema of the system is necessary so as to know and represent data from the viewpoint of the user, independent of any technology that will be used to implement the model.

The following are the outlines for the development of the Conceptual Schema:

- Describe the entities.
- Describe and justify the relations among the entities.
- Define the relations.
- Describe the structural constraints of the scenario in detail.
- Develop the enhanced entity relationship model.

Verify the EER Model hence developed.

3.1 Entities Definition

Entities are the principal data object about which information is to be collected. Entities are usually recognizable concepts, either concrete or abstract, such as person, places, things, or events, which have relevance to the database. Entities are classified as regular or weak. A regular entity is one that does not rely on another for identification. A weak entity is one that relies on another for identification. An entity occurrence (also called an instance) is an individual occurrence of an entity. An occurrence is analogous to a row in the relational table.

ENTITY

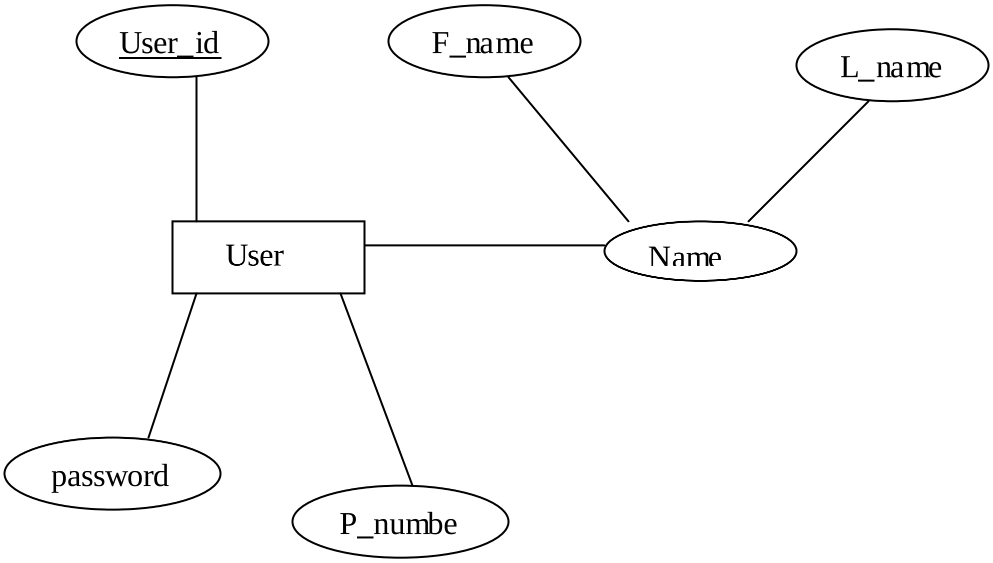
Entities are principal data object about which information is to be collected. Entities are usually recognizable concepts, either concrete or abstract, such as person, places, things, or events which have relevance to the databases

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

In our project the entities are seven as follows:

1. User
2. Registration
3. Update test
4. Choose tech
5. Result

USER:

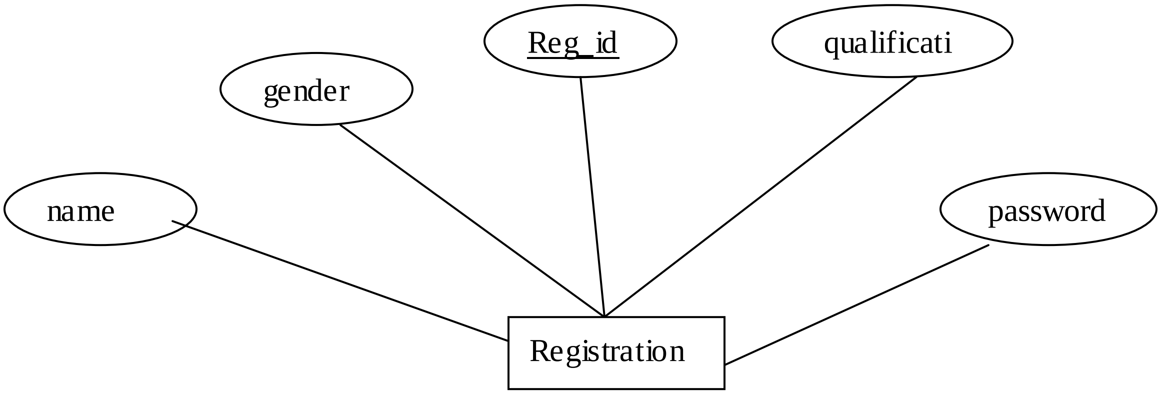


Type: Regular	
Identifier : USER:	

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Attribute	**Description**
User_id	User 's identification(Primary key)
Name	User 's name
Password	User password
Phone No	User 's phone number

Registration

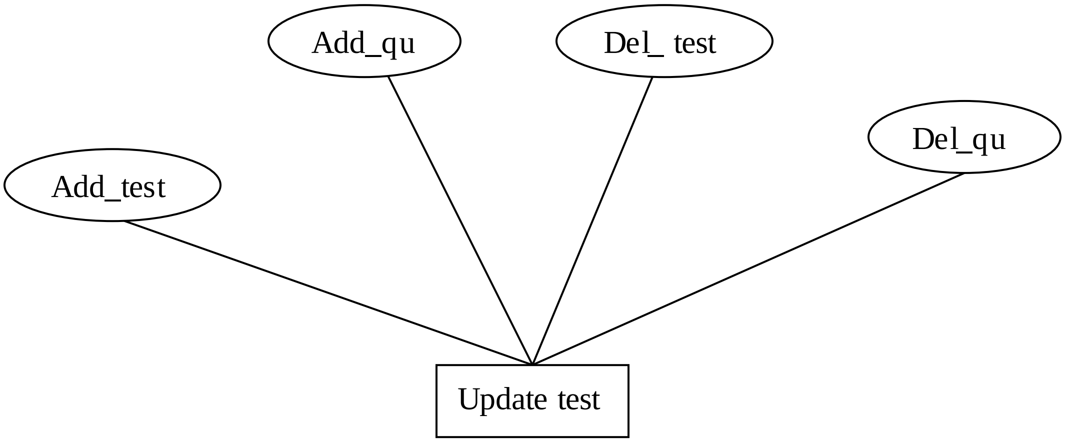


Type: Regular

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Identifier : Registration	
Attribute	**Description**
Name	User 's name
Password	User password
Gender	User gender
Reg_id	User Reg_id(Primary key)
Qualification	User qualification

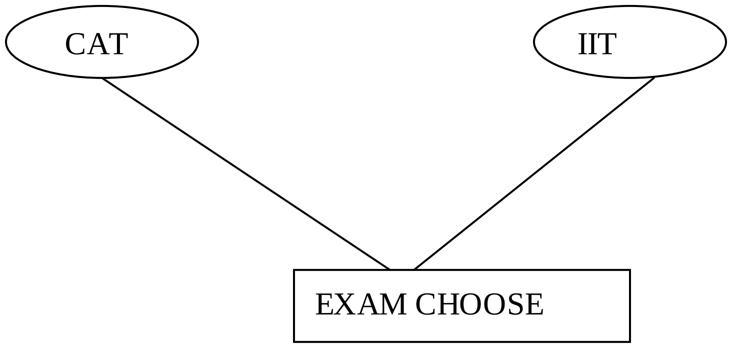
Update test:-



Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Type: Regular	
Identifier : Update test	
Attribute	**Description**
Add test	Add test according to user
Add_question	Add question in test
Del_ test	Delete test
Del_qu	Delete question

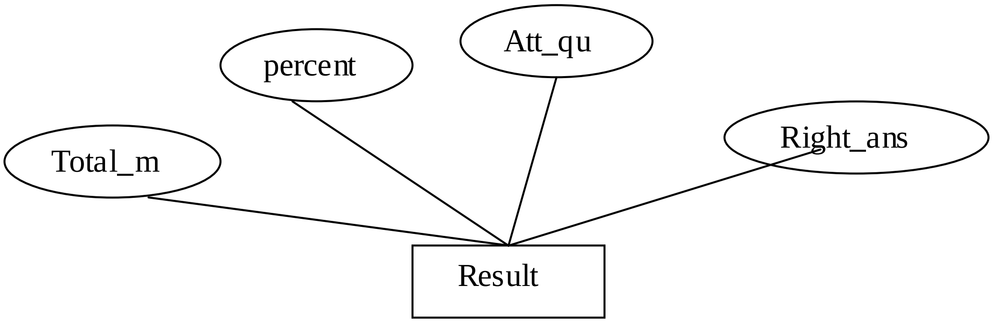
Choose tech:-



Type: Regular	
Identifier : Choose Exam	
Attribute	**Description**
Cat	Management study
IIT	Engineering exam

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Result



Type: Regular	
Identifier : Result	
Attribute	**Description**
Total_m	Total marks of given test

3.2 Relationship among Entities

Relationship is an association among the instances of one or more entity types that is of interest to the organization. A relationship type is a meaningful association between entity types. Relationships are classified in terms of degree, connectivity, cardinality, and existence. There are three degrees of relations - unary, binary and Ternary.

Unary relationships are relationships between the instances of single entity types. The business model chosen does not contain any unary relationship.

Binary relationships are relationships between instances of two entity types. There can be three types of binary relationships – one to one, one to many and many to many categorized on the basis of cardinalities and participation.

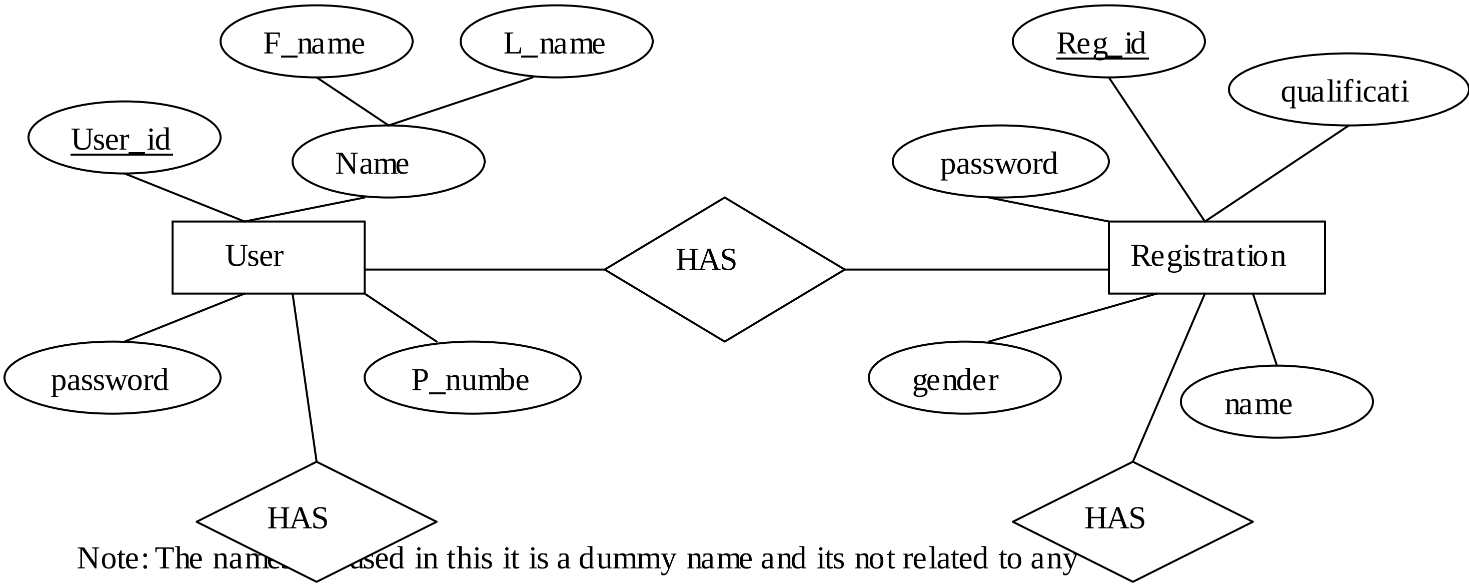
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Cardinality specifies the number of instances of one entity that can be associated with each instance of another entity.

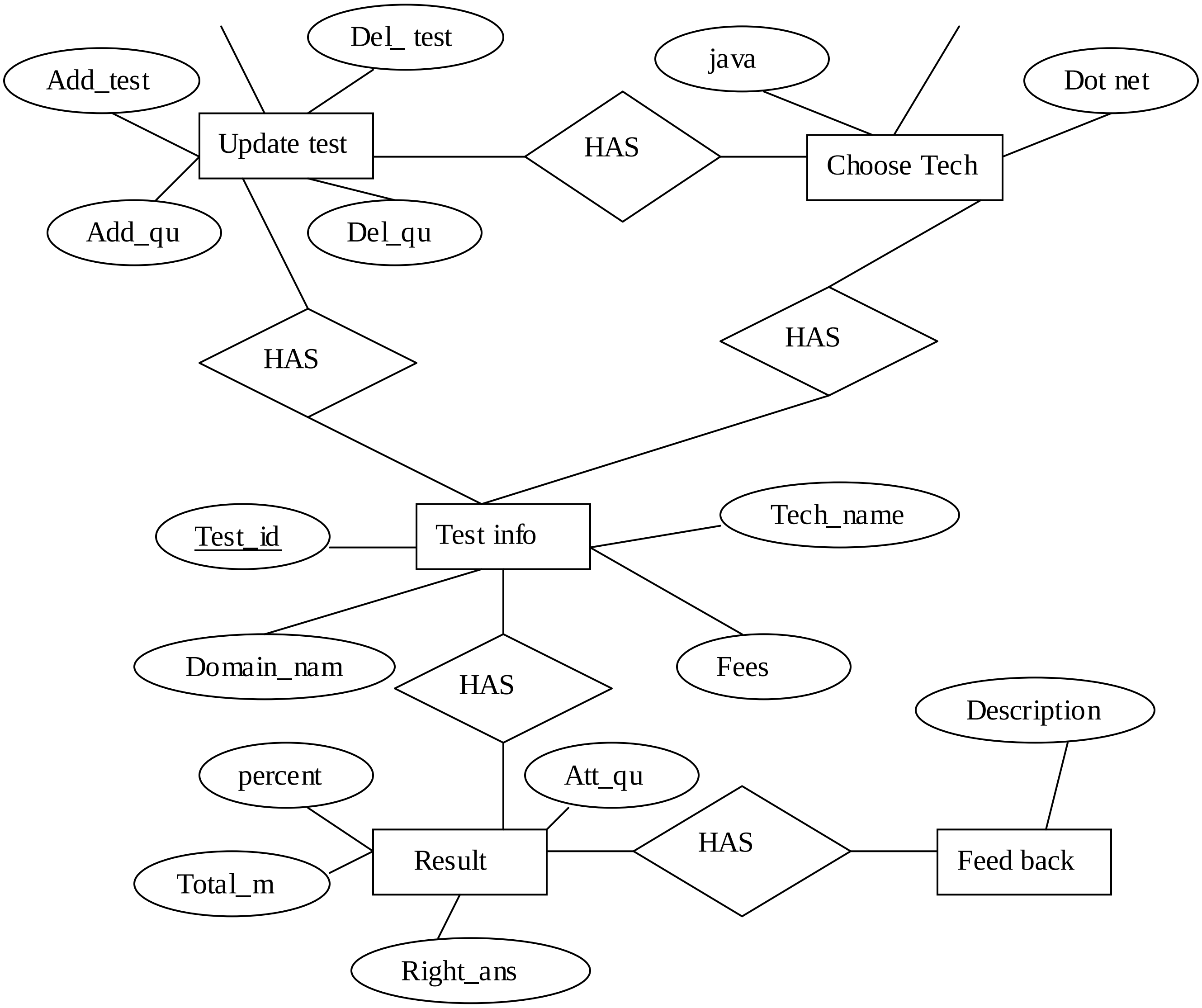
Participation is the minimum cardinality of a relationship.

The relationship among the various entities can be shown in the form of a table. It is also possible that an entity may not have any relation with some other entity.

3.3 E-R diagram



Note: The name used in this it is a dummy name and its not related to any company or college or any person.



3.4 Entity Relationship Model

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.

Note: The names we used in this it is a dummy name and its not related to any
company or college or any person.

- It is simple and easy to understand with a minimum of training. Therefore, the database designer to communicate the design to the end user can use model.

The model can be used as a design plan by the database developer to implement a data model in specific database management software.

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation.

- Entities are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- A solid line connecting two entities represents relationships. The name of the relationship is written above the line. Relationship names should be verbs.
- Attributes, when included, are listed inside the entity rectangle. Attributes, which are identifiers, are underlined. Attribute names should be singular nouns.
- A line ending in a crow’s foot represents cardinality of many. If the crow's foot is omitted, the cardinality is one.

3.5 Logical Design

As of now, the conceptual data model of the system is prepared. Now the next is to prepare the logical schema of the model, which is presented in this chapter. In this chapter, you have to transform conceptual schema into a standard relational model. Logical Database design is the process of transforming the Conceptual Data Model into a logical data model. The Relational Data Model is the most commonly used in contemporary database applications. It also has a wide tool support. The objective of logical model is to translate the conceptual design (which represents the organization requirements) into a logical design that can be implemented as an information system.

The resulting databases must meet the users' needs for data sharing, flexibility and ease of access.

The following are the objectives of Logical Schema:

- Analyze in detail the transactions, forms, and displays required by the business functions and supported by the database.
- Transformation of Conceptual Data Model into a Relational Model with data specifications.
- Transformation of data specifications into basic elements for well-structured relations and data specifications (Normalization).

The objective of logical database design is to translate the conceptual design into a logical database design like a Relational data model. The Relational data model represents data in the form of tables. Its components are -

- Data Structure: Data Organized in the form of rows and columns.
- Data Manipulation: Powerful operations (SQL) are used to manipulate data stored in relations.
- Data Integrity: Facilities are included to specify business rules that maintain the integrity of data when they are manipulated.
- A relation is thus a named two-dimensional table of data. Each relation (or table) consists of a set of named columns and an arbitrary number of unnamed rows.

Properties of a relation

- Each relation in a database has a unique name.
- An entry at the intersection of each row and column is atomic. There can be no multi-valued attributes in a relation.
- Each row is unique. No two rows in a relation are identical.
- Each attribute (or column) within a table has a unique name.

- The sequence of column and row is insignificant.

3.6 Transforming ER Diagrams into Relational Schema

The input to this process is the Enhanced Entity Relationship Model. The outputs are the relational schemas. Transforming the EER diagram into relations is a relatively straightforward process with a well-defined set of rules. The following steps show the conversion of the ER Diagram to the relations.

<u>Tables</u>: There are no of table uses for different purpose are as follows:

USERS-TABLE

Attributes	Data type	Constraints (Validation)	Description
UID	Varchar(20)	Primary Key	User id for identification

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

			OF USER
User Name	Varchar(50)	Not Null	Name of User
Password	Varchar(20)	Not Null	Password for Security
Date of Birth	Date Time(8)	Not Null	Date of Birth of user
Gender	Varchar (10)	Not Null	Specify the user is male or female
Professional Qualification	Varchar(20)	Not Null	Professional Qualification of user
University/Institutes	Varchar(50)	Not Null	University name or institutes name where he/she got Education
Educational Qualification	Varchar(20)	Not Null	Qualification of user
University/Institutes	Varchar(50)	Not Null	University name or institutes name where he/she got Education
Phone No	Varchar (20)	Not Null	Phone number of user with std code
Mob No	Varchar(20)	Null	Mobile number of user
E-Mail	Varchar(50)	Null	E-mail address of user
Address	Varchar(100)	Not Null	Address of user
City	Varchar(20)	Not Null	City Name Where user give the test
Zip Code	Varchar(10)	Not Null	Zip code of city where user giving test
State	Varchar(50)	Not Null	State Which Have user Belong
Country	Varchar(25)	Not Null	Country where user live .

Question table:

Attributes	Data type	Constraints (Validation)	Description
Question No	Int(4)	Identity Column	Question number of question sheet for related technology .It is

			automatically created by System.
Question	Varchar(100)	Not Null	Questions according to Test ID.
Option1	Varchar(20)	Not Null	Option 1 for objective type of answer
Option2	Varchar(20)	Not Null	Option 2 for objective type of answer
Option3	Varchar(20)	Not Null	Option 3 for objective type of answer
Option4	Varchar(20)	Not Null	Option 4 for objective type of answer
Answer	Int(4)	Not Null	Answer number of according to question number for related Test ID.
Test Id	Varchar(20)	Not Null	Test ID According to Questions.

CHAPTER 4
User Interface Design

The chapter shows the process of creating user interfaces for the desired system. Before creating a software interface, all the transaction and input/output must be known in advance and it should be clearly stated. The chapter also presents the design of the user interface i.e. how the interface is organized. In the last, the implementation of the interface is shown.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Interface is the medium through which the user communicates with the application. The User Interface is one of the most important parts of software. The User Interface acts as a mediator between the user and the system. It is important, that the user should feel comfortable while using the software. Software is considered as unsuccessful if it does not gain the user acceptance. Although however good the software is but if the user finds it difficult to use it then the user will definitely reject the software, and the software will be a failure. So besides having good features, a software must also have a user friendly Interface.

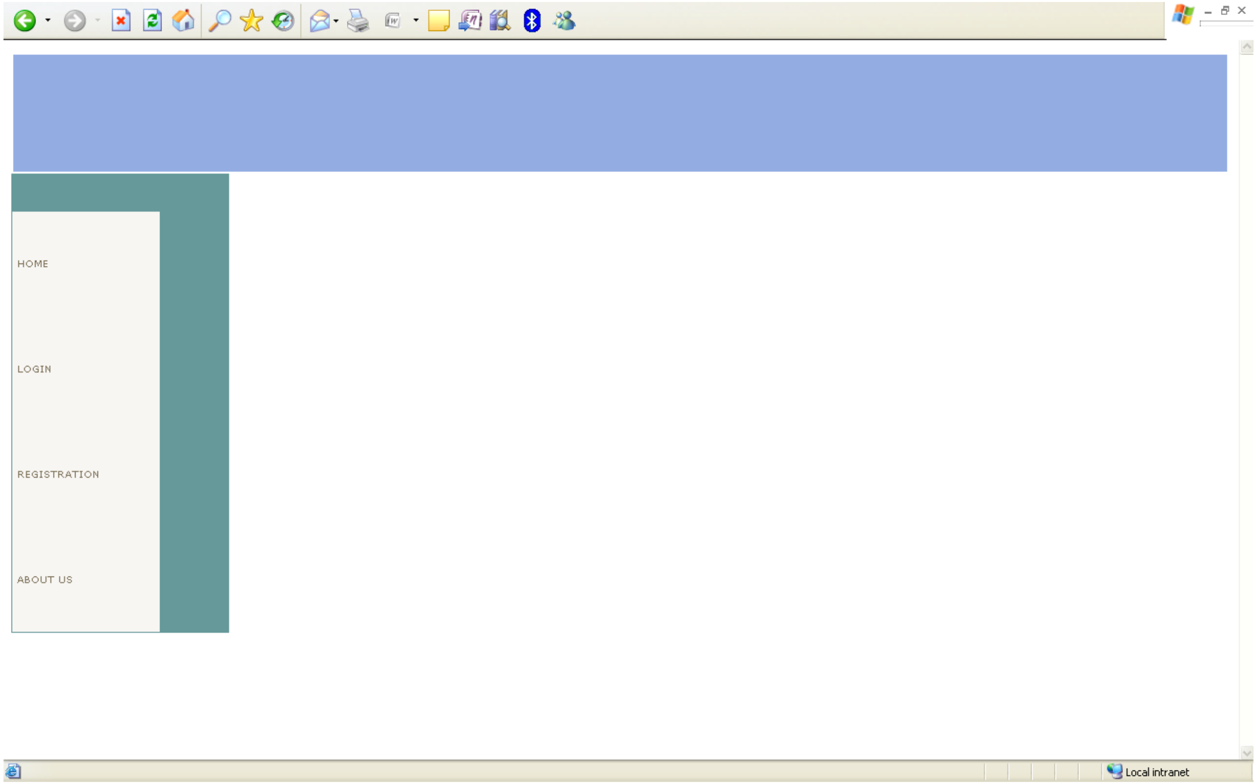
The Objectives behind creating a User Interface are as follows:

- The Interface must provide all the features, which are discussed earlier.
- Besides providing features, the interface must also be user friendly.

Here we have pasted the picture of Form and Report developed by us.

Main Form of the Project:

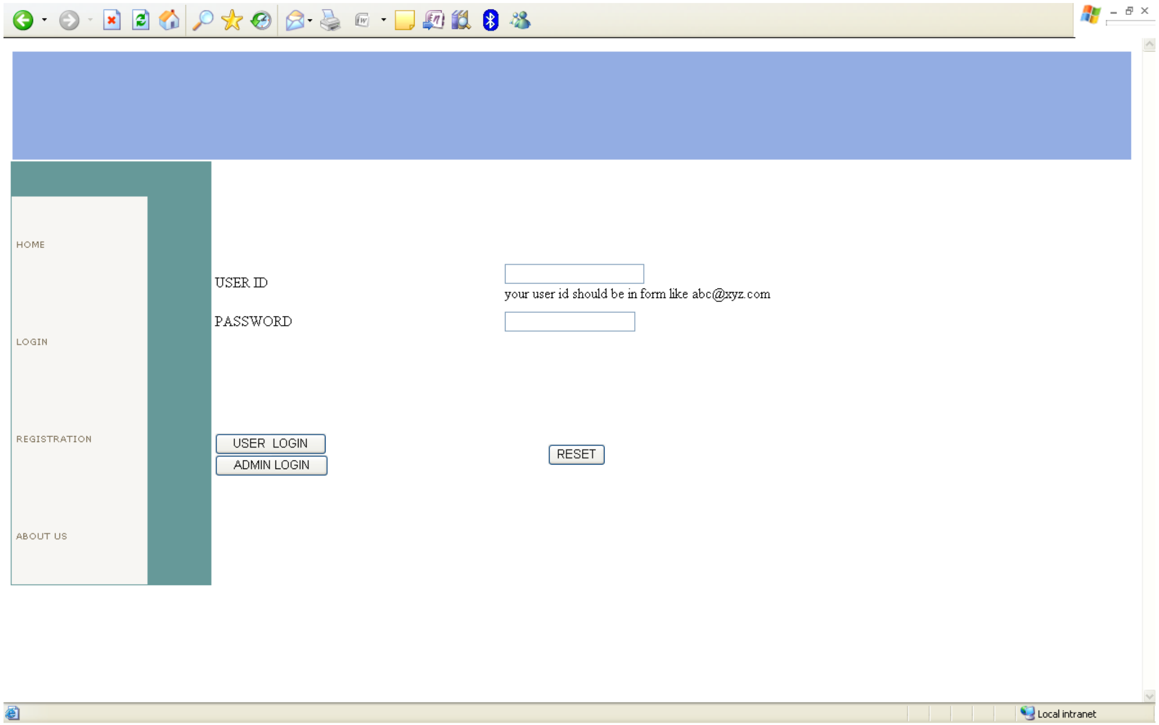
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.



Purpose	To login for the Test.
Inputs	Enter User name and Password.
Processing	Check the user name and password
Outputs	Go to the next page according to user.

Administrator Login Page:

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.



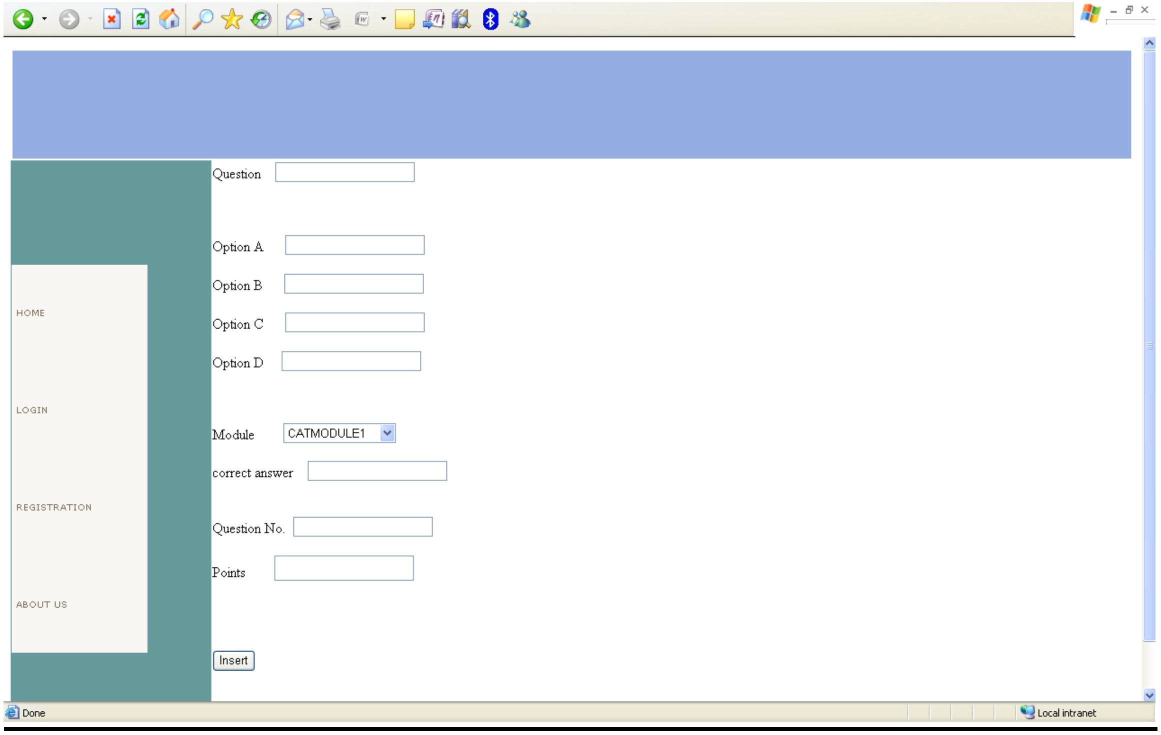
USER ID

your user id should be in form like abc@xyz.com

PASSWORD

Add Question

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

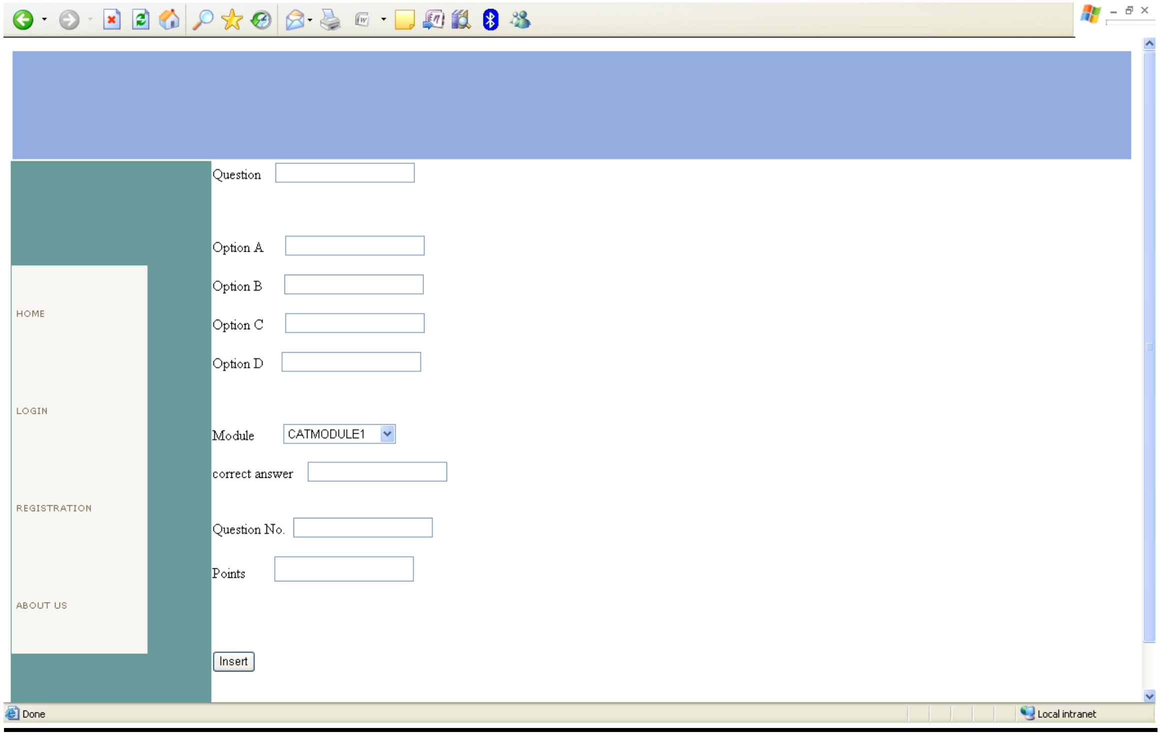


Module CATMODULE1

Purpose	To add Question in paper.
Inputs	Question
Processing	Add the question
Outputs	New Question will be add.

Update Question:

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.



Question

Option A

Option B

Option C

Option D

Module CATMODULE1

correct answer

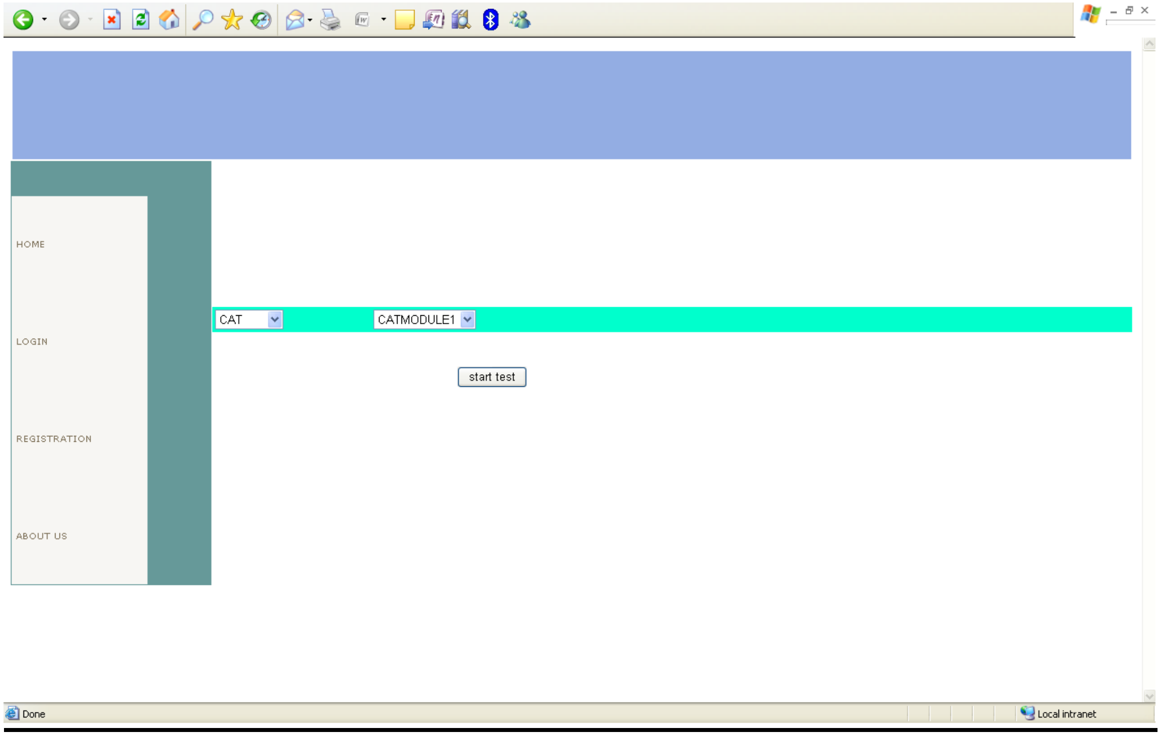
Question No.

Points

Purpose	To Update the Question.
Inputs	Question
Processing	Update of question
Outputs	Question is update

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

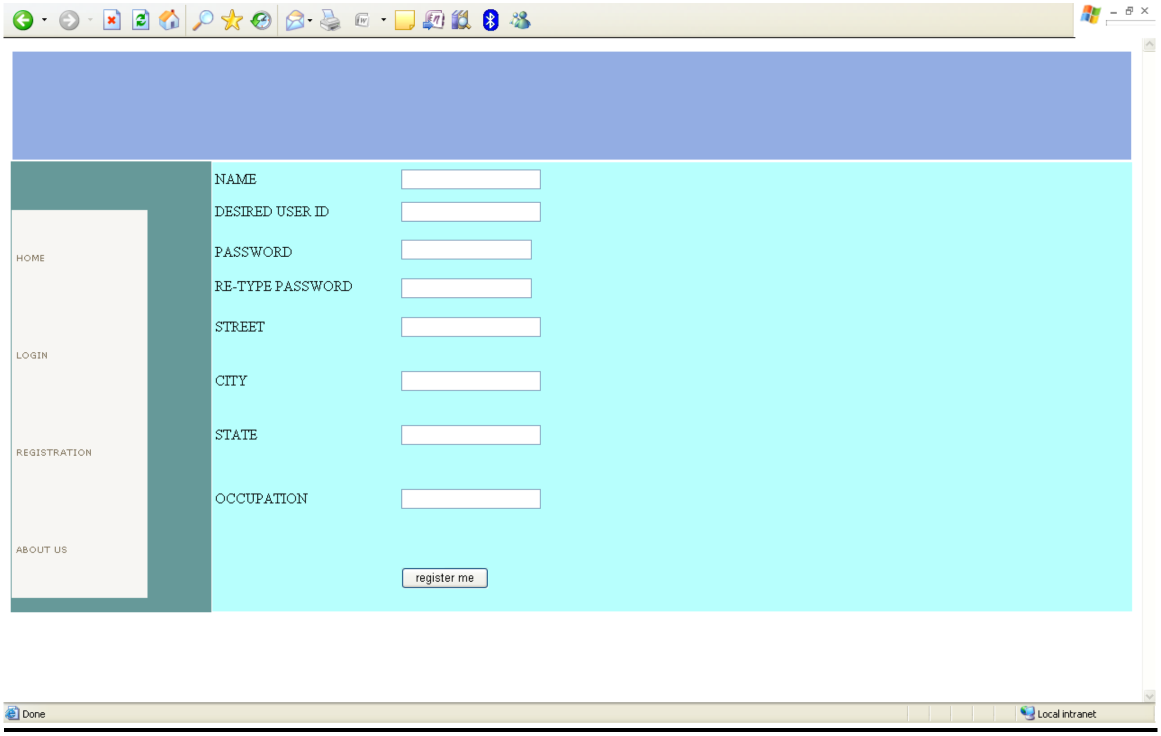
Selection Sample Test Page:



Purpose	Select sample paper.
Inputs	Select appropriate paper.
Processing	Select paper.
Outputs	Sample paper selected.

Registration Form:

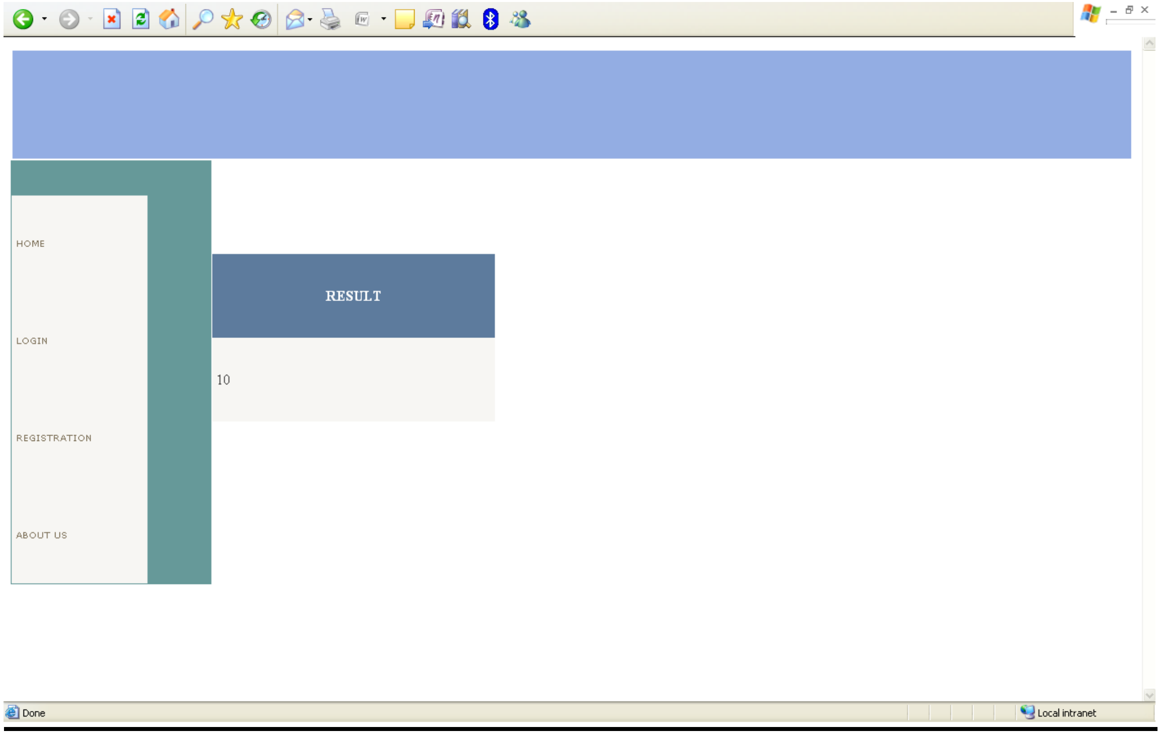
Note: The names we used in this it is a dummy name and its not related to any company or college or any person.



Purpose	Register the user.
Inputs	Fill all entry
Processing	Select all appropriate answer
Outputs	Your form is fill.

Display Result:

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.



RESULT

10

Purpose	Show the result
Inputs	Enrollment number
Processing	result
Outputs	Show your test result

CHAPTER 5

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

CODING

Login page

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using
System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class login :
System.Web.UI.Page
{
    SqlConnection con = new
SqlConnection("data
source=.;database=kuber;trusted_connect
ion=yes");
    protected void Page_Load(object
sender, EventArgs e)
    {
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
    }
    protected void Button1_Click(object
sender, EventArgs e)
    {
      SqlCommand cmd = new
SqlCommand("select * from login1 where
userid='" + txblouserid.Text + "' and
password=" + txblopassword.Text, con);
        //SqlCommand cmd1 = new
SqlCommand("select name from login1
where
userid='''+txblouserid.Text+"''",con);
        con.Open();
       // SqlDataReader dr1 =
cmd.ExecuteReader();
        SqlDataReader dr =
cmd.ExecuteReader();
        if (dr.Read())
        {
            txtsesion.Text =
dr[0].ToString();
            Session["name"] =
txtsesion.Text;
            Response.Redirect("exam
info.aspx");
           }
        else
            Response.Redirect("error
page.aspx");
```

```
        //SqlCommand cmd1=new
SqlCommand ("select name from login1
where userid='"+txblouserid+"'",con);
        //con.Open();
        //SqlDataReader
dr1=cmd1.ExecuteReader();

        //if (dr1.Read())
        //{
        //    txtsesion.Text =
dr1[0].ToString();
        //    Session["name"] =
txtsesion.ToString();

        //}

        con.Close();
    }
    protected void
btnreset_Click(object sender, EventArgs
e)
    {
        txblouserid.Text = "";
        txblopassword.Text = "";
    }
    protected void Button2_Click(object
sender, EventArgs e)
    {
        if (txblouserid.Text ==
"admin@admin.com" && txblopassword.Text
== "admin")
```

```
        {
Response.Redirect("~/QADD1.aspx");
        }
        else
            Response.Write("wrong
password or id");
    }
}
```

Registration page

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
<%@ Page Language="C#"
MasterPageFile="~/firstMasterPage.maste
r" AutoEventWireup="true"
CodeFile="registration.aspx.cs"
Inherits="loginpage" Title="Untitled
Page" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="ContentPlaceHolde
r1" Runat="Server">
    <table style="width: 100%; height:
499px; background-color: #B7FFFD">
    <tr>
        <td class="style3"
style="width: 203px">
            <asp:Label ID="Label1"
runat="server" Text="NAME"></asp:Label>
        </td>
        <td>
            <asp:TextBox ID="txbnam"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator1"
runat="server"

ControlToValidate="txbnam"
Display="Dynamic"

ErrorMessage="RequiredFieldValidator">p
lease enter your
name</asp:RequiredFieldValidator>
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
        </td>
    </tr>
    <tr>
        <td class="style3"
style="width: 203px">
            <asp:Label ID="Label2"
runat="server" Text=" DESIRED USER
ID"></asp:Label>
        </td>
        <td>
            <asp:TextBox ID="txbuserid"
runat="server"></asp:TextBox>

<asp:RegularExpressionValidator
ID="RegularExpressionValidator1"
runat="server"

ControlToValidate="txbuserid"
Display="Dynamic"

ErrorMessage="RegularExpressionValidato
r"

ValidationExpression="\w+([-
+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*">enter
            your valid email
address</asp:RegularExpressionValidator
>
```

```
            <asp:Label ID="lbluserid" 
runat="server" Font-Bold="True" Font-
Italic="True" 
                Font-Names="Courier 
New" ForeColor="#000066"></asp:Label>
        </td>
    </tr>
    <tr>
        <td class="style3" 
style="width: 203px">
            <asp:Label ID="Label3" 
runat="server" 
Text="PASSWORD"></asp:Label>
            <br />
            <br />
            <asp:Label ID="Label8" 
runat="server" Text="RE-TYPE 
PASSWORD"></asp:Label>
        </td>
        <td>
            <asp:TextBox ID="txbpass" 
runat="server" 
TextMode="Password"></asp:TextBox>
            <asp:RequiredFieldValidator 
ID="RequiredFieldValidator2" 
runat="server" 

ErrorMessage="RequiredFieldValidator" 
ControlToValidate="txbpass" 
                Display="Dynamic">enter 
password</asp:RequiredFieldValidator>
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
            <br />
            <br />
            <asp:TextBox ID="TXBRETYPE" 
runat="server" 
TextMode="Password"></asp:TextBox>
            <asp:CompareValidator 
ID="CompareValidator1" runat="server" 

ControlToCompare="txbpass" 
ControlToValidate="TXBRETYPE" 
Display="Dynamic" 

ErrorMessage="CompareValidator" 
Type="Integer">check 
password</asp:CompareValidator>
        </td>
    </tr>
    <tr>
        <td class="style3" 
style="width: 203px">
            <asp:Label ID="Label4" 
runat="server" 
Text="STREET"></asp:Label>
        </td>
        <td>
            <asp:TextBox ID="txbstreet" 
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator 
ID="RequiredFieldValidator3" 
runat="server" 
```

```
ControlToValidate="txbstreet"
Display="Dynamic"

ErrorMessage="RequiredFieldValidator">e
nter
street</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td class="style3"
style="width: 203px; height: 61px">
            <asp:Label ID="Label5"
runat="server" Text="CITY"></asp:Label>
        </td>
        <td style="height: 61px">
            <asp:TextBox ID="txbcity"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator4"
runat="server"

ControlToValidate="txbcity"
Display="Dynamic"

ErrorMessage="RequiredFieldValidator">e
nter city</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
```

```
        <td class="style3"
style="width: 203px">
            <asp:Label ID="Label6"
runat="server"
Text="STATE"></asp:Label>
        </td>
        <td>
            <asp:TextBox ID="txbstate"
runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator5"
runat="server"

ControlToValidate="txbstate"
Display="Dynamic"

ErrorMessage="RequiredFieldValidator">e
nter state</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td style="height: 78px; width:
203px">
            <asp:Label ID="Label7"
runat="server"
Text="OCCUPATION"></asp:Label>
        </td>
        <td style="height: 78px">
            <asp:TextBox ID="txboccu"
runat="server"></asp:TextBox>
```

```
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator6"
runat="server"

ControlToValidate="txboccu"
Display="Dynamic"

ErrorMessage="RequiredFieldValidator">e
nter
occupation</asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td style="height: 51px; width:
203px">
        </td>
        <td style="height: 51px">
            <asp:Button
ID="BTNREGISTER" runat="server"
onclick="BTNREGISTER_Click"
                Text="register me" />
        </td>
    </tr>
</table>
</asp:Content>
```

Add Question

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/firstMasterPage.maste
```

```
r" AutoEventWireup="true"
CodeFile="QADD1.aspx.cs"
Inherits="QADD1" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="ContentPlaceHolde
r1" Runat="Server">
    <asp:Label ID="Label1"
runat="server"
Text="Question"></asp:Label>
&nbsp;&nbsp;
    <asp:TextBox ID="txtqsn"
runat="server"></asp:TextBox>
    <br />
    <br />
    <br />
    <br />
    <asp:Label ID="Label2"
runat="server" Text="Option
A"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop1"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label3"
runat="server" Text="Option
B"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop2"
runat="server"></asp:TextBox>
```

```
    <br />
    <br />
    <asp:Label ID="Label4"
runat="server" Text="Option
C"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop3"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label5"
runat="server" Text="Option
D"></asp:Label>
&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop4"
runat="server"></asp:TextBox>
    <br />
    <br />
    <br />
    <br />
    <asp:Label ID="Label6"
runat="server"
Text="Module"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:DropDownList
ID="DropDownList1" runat="server"
AutoPostBack="True">

<asp:ListItem>CATMODULE1</asp:ListItem>

<asp:ListItem>CATMODULE2</asp:ListItem>
```

```
<asp:ListItem>CATMODULE3</asp:ListItem>

<asp:ListItem>CATMODULE4</asp:ListItem>

<asp:ListItem>IITMODULE1</asp:ListItem>

<asp:ListItem>IITMODULE2</asp:ListItem>

<asp:ListItem>IITMODULE3</asp:ListItem>

<asp:ListItem>IITMODULE4</asp:ListItem>

<asp:ListItem>GATEMODULE1</asp:ListItem
>

<asp:ListItem>GATEMODULE2</asp:ListItem
>

<asp:ListItem>GATEMODULE3</asp:ListItem
>

<asp:ListItem>GATEMODULE4</asp:ListItem
>

<asp:ListItem>CBSEMODULE1</asp:ListItem
>

<asp:ListItem>CBSEMODULE2</asp:ListItem
>
```

```
<asp:ListItem>CBSEMODULE3</asp:ListItem
>

<asp:ListItem>CBSEMODULE4</asp:ListItem
>
        <asp:ListItem></asp:ListItem>
    </asp:DropDownList>
    <br />
    <br />
    <asp:Label ID="Label9"
runat="server" Text="correct
answer"></asp:Label>
&nbsp;&nbsp;
    <asp:TextBox ID="txtans"
runat="server"></asp:TextBox>
    <br />
    <br />
    <br />
    <asp:Label ID="Label7"
runat="server" Text="Question
No."></asp:Label>
&nbsp;<asp:TextBox ID="txtqno"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label8"
runat="server"
Text="Points"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
```

```
    <asp:TextBox ID="txtpoints" 
runat="server" 
Height="22px"></asp:TextBox>
    <br />
    <br />
    <br />
    <br />
    <br />
    <asp:Button ID="Button1" 
runat="server" onclick="Button1_Click" 
Text="Insert" />
    <br />
    <br />
    <br />
    <br />
</asp:Content>
```

Main Exam

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/firstMasterPage.maste
r" AutoEventWireup="true"
CodeFile="QADD1.aspx.cs"
Inherits="QADD1" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="ContentPlaceHolde
r1" Runat="Server">
    <asp:Label ID="Label1"
runat="server"
Text="Question"></asp:Label>
&nbsp;&nbsp;
    <asp:TextBox ID="txtqsn"
runat="server"></asp:TextBox>
    <br />
    <br />
    <br />
    <br />
    <asp:Label ID="Label2"
runat="server" Text="Option
A"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop1"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label3"
runat="server" Text="Option
B"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
    <asp:TextBox ID="txtop2"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label4"
runat="server" Text="Option
C"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop3"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label5"
runat="server" Text="Option
D"></asp:Label>
&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtop4"
runat="server"></asp:TextBox>
    <br />
    <br />
    <br />
    <br />
    <asp:Label ID="Label6"
runat="server"
Text="Module"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:DropDownList
ID="DropDownList1" runat="server"
AutoPostBack="True">

<asp:ListItem>CATMODULE1</asp:ListItem>
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
<asp:ListItem>CATMODULE2</asp:ListItem>

<asp:ListItem>CATMODULE3</asp:ListItem>

<asp:ListItem>CATMODULE4</asp:ListItem>

<asp:ListItem>IITMODULE1</asp:ListItem>

<asp:ListItem>IITMODULE2</asp:ListItem>

<asp:ListItem>IITMODULE3</asp:ListItem>

<asp:ListItem>IITMODULE4</asp:ListItem>

<asp:ListItem>GATEMODULE1</asp:ListItem>

<asp:ListItem>GATEMODULE2</asp:ListItem>

<asp:ListItem>GATEMODULE3</asp:ListItem>

<asp:ListItem>GATEMODULE4</asp:ListItem>

<asp:ListItem>CBSEMODULE1</asp:ListItem>
```

```
<asp:ListItem>CBSEMODULE2</asp:ListItem
>

<asp:ListItem>CBSEMODULE3</asp:ListItem
>

<asp:ListItem>CBSEMODULE4</asp:ListItem
>
        <asp:ListItem></asp:ListItem>
    </asp:DropDownList>
    <br />
    <br />
    <asp:Label ID="Label9"
runat="server" Text="correct
answer"></asp:Label>
&nbsp;&nbsp;
    <asp:TextBox ID="txtans"
runat="server"></asp:TextBox>
    <br />
    <br />
    <br />
    <asp:Label ID="Label7"
runat="server" Text="Question
No."></asp:Label>
&nbsp;<asp:TextBox ID="txtqno"
runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Label ID="Label8"
runat="server"
Text="Points"></asp:Label>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    <asp:TextBox ID="txtpoints"
runat="server"
Height="22px"></asp:TextBox>
    <br />
    <br />
    <br />
    <br />
    <br />
    <asp:Button ID="Button1"
runat="server" onclick="Button1_Click"
Text="Insert" />
    <br />
    <br />
    <br />
    <br />
</asp:Content>
```

Result

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
public partial class newtech1result :
System.Web.UI.Page
{ string username;

    SqlConnection con = new
SqlConnection("data
source=.;database=kuber;trusted_connect
ion=yes");
    DataSet ds=new DataSet();
    protected void Page_Load(object
sender, EventArgs e)
    {
        if (Session["name"] == null)
        {

Response.Redirect("~/home.aspx");
        }
        username =
Session["name"].ToString();
        string con;
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
        con = "data
source=.;database=kuber;trusted_connect
ion=yes";
        string command;
        //command="select  uans as
'your answer'  ,ans as 'correct answer'
,point from "+username +"";
        //command = "select sum(point)
as 'RESULT' from ' username' where ans
= uans";
//        SqlCommand cmd = new
SqlCommand();
        SqlDataAdapter da = new
SqlDataAdapter("select sum(point) as
'RESULT' from " + username + " where
ans = uans ", con);
        // SqlDataAdapter da = new
SqlDataAdapter(cmd,con);
        da.Fill(ds);

GridView1.DataSource=ds.Tables[0];
       GridView1 .DataBind();

    }
    protected void
GridView1_SelectedIndexChanged(object
sender, EventArgs e)
    {

    }
```

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

```
    protected void 
GridView1_DataBound(object sender, 
EventArgs e)
    {
        con.Open();
        SqlCommand cmd = new 
SqlCommand();
        cmd.CommandText = "drop table " 
+ username;
        cmd.Connection = con;
        cmd.ExecuteNonQuery();
        con.Close();
    }
}
```

CHAPTER 6

TESTING

5.1 INTRODUCTION

Till now the database design, user interface design and implementation are complete. The system now is tested for its functionality, validity and performance. In order to test the system, a wide variety of tests are conducted to make sure that the system matches the entire identified user requirements and constraints. This chapter focuses on testing the developed systems using different test strategies in order to verify its correctness and user acceptance.

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as yet undiscovered error. A successful test is one that uncovers an as yet undiscovered error.

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

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. And it needs to be done in almost every phase of product development life cycle not just before a product is handed to a customer.

Note: The names we used in this it is a dummy name and its not related to any
company or college or any person.

The following are some attributes of a good test:

- A good test has a high probability of finding an error. To achieve this goal the tester must understand the software and attempt to develop a mental picture of how the software may fail. Ideally the classes of failure are probed.
- A good test is not redundant: testing time and resources are limited. There is no point in conducting the test that has the same purpose as another test. Every test should have a different purpose.
- A good test should be best of breed. In a group of tests that have a similar intent time and resource limitations may militate for the execution of only a subset of these tests. In such cases the tester that has the highest likelihood of uncovering a whole class of errors should be used.
- A good test should be neither too simple nor too complex: although it is sometimes possible to combine a series of tests into one test case, the possible side effects associated with this approach may mask errors. In general each test should be executed separately.

5.2 TYPES OF TESTING

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit software design- the module. Using the procedural design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in algorithmic execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths (bases paths) through the control structure are exercised to ensure that all elements in a module have been executed at least once. And finally all error-handling paths are tested.

Application interface of our system was unit tested at all levels of implementation, right from start of code writing, to integrating the code with other modules. Every module was tested fully to check its syntax and logical correctness. Error handling was implemented into relevant modules so that the code doesn't crash on errors.

5.2.2 Integration Testing

Integration testing is a systematic technique for constructing the program structures, while conducting test to uncover errors associated with interfacing, the objective is to take unit tested modules and build a program structure that has been dictated by design.

User interface of i-Admit was developed in modules. All of them were joined together to make the complete running application. While integrating these modules, integration testing was performed on them to verify that they meet all interfacing requirements and that they pass relevant information among themselves. In the end the complete program structure was tested to ensure interoperability of all the modules.

5.2.3 Validation Testing

At the culmination of integration testing software is completely assembled as a package: interfacing errors have been uncovered and corrected and a final series of software tests – Validation Testing may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of Black Box tests that demonstrate conformity with requirements.

5.2.4 Alpha Testing

It is virtually impossible for a software developer to foresee how the customer will really use a program. When custom software is built for one customer a series of acceptance tests are conducted to enable the customer to validate all requirements.A customer conducts the alpha test at the developer site. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problem. Alpha tests are conducted in a controlled environment. Alpha tests were performed at our development site with the help of our friends, who were called and asked to run the program in the manner they like, without our guidance and errors and usage problems were noted and code was updated to remove all of them.

5.2.5 Beta Testing

The Beta test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing the developer is generally not present. Therefore the beta test is a live application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta test the software developer makes modification and then prepares for the release of software product to the entire customer base.Beta testing of our system is not performed as the product is not yet fully developed and has not been installed at the user site as it still is in the development phase. Beta testing will be performed when the software is deployed at the user's site.

5.2.6 System Testing

Ultimately software is incorporated with other system elements (new hardware, information) and a series of system, integration and validation tests are conducted. It is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose all work to verify that all system elements have been properly integrated and perform allocated functions.System testing of this system was performed at the development lab of this system by integrating the functional systems to imitate the actual work environment.

.

5.2.7 Security Testing

Security Testing attempts to verify protection mechanism built into a system will in fact protect it from improper penetration. Security is provided for each user by giving them login name and password. Security testing was done, as any other anonymous user can’t log in with a user password if the user is already logged in.

5.3 Test Plan:

A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning

Cases	Input	Problem	Expected Output
General bugs			
User name	User name	Not working	Done
password	Password	Not working	Done
Adding technology			
Test _id	Give test id	Not working	Done
Technology name	Given Technology name According to user	Not working	Done
Domian name	Given Domain name According to user	Not Working	Done
Delete user			
Delete existing user all record	In this from delete existing user all record	Not Working	Done
Adding questions			
In this from the administrator user login add add the question time to time .	In this from the administrator user login add add the question time to time .	Not Working	Done
In this from the administrator user given the number of four option	In this from the administrator user given the number of four option	Not Working	Done

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Chosen one option	Chosen one option		
Answer	Select appropriate answer	Not Working	Done
TestId	Give the test id by administrator	Wrong	Done
Delete			Done
Edit and Update			Done

Registration

Uid, Name, Password, Dob Phone No., Email ID, Qualification	Submit all detail of user	Major	Displayed

Result

Test ID	ID of given Test	Critical	Done
Attempt Question	Number of attempt question	Major	Done
Total Marks	Marks you get	Cosmetic	Done

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Chapter-7

Discussion & Conclusions

6.1 CONCLUSION

The project seems to be quiet satisfactory as it is fulfilling the requirement of end-user's it is live project implemented in the organization. And working well in the organization.

6.2 LIMITATION:

Everything in this world irrespective of its walk of life has its pros &cons. There is any dark side to any work so as to compensate the good side of it. Thus the limitation is it doesn't contain employee payroll system and for that there is another method i.e. it is maintained through EXCEL sheet

Where the daily attendance of employee is recorded. For this the end user has to maintain different registers where the concerning data should be kept

6.3 LESSONS LEARNED:

As it happens with everyone, whenever a person ventures in something new he/she is bound to falter at some stage. Same is happened with us And here we learn some lesson during the development of this project and is if you try your best you will definitely get success .In starting during Information gathering time, we were not that much efficient that we understand the user requirement clearly. Thus we face many problems like after designing the database, there is some change in the user requirement thus we designed the database again. And the name given by us (abbrev.) Is also some where not understood by the end user. And there is change in the output format required by the end user i.e. there are some fields which must be displayed in the output form but not in the report form.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Thus a software requirement specification document must be prepared in starting where the requirement of the project is described otherwise gathering of information again and again made the work

CHAPTER 8

References

Appendix A

Here the Project Work log is defined for our project

S.no	Team Members---→ Work done	Amit	Ashish	Sachin	Rahul
1.	Problem Analysis	***	**	*	*
2.	Requirement analysis And Design	***	***	*	*
3.	Database Design	**	*	**	**
4.	Form & Report Design	*	***	***	***
5.	Code Writing	*	*	***	***
6.	Testing	**	***	**	**
7.	Documentation	***	***	*	*

❖ Professional C#.Net 2005:Wrox Press
❖ Visual C# .NET and the .NET Platform: An Advance Guide
❖ Asp.Net E-Books.
❖ System Analysis and Design Methods: IGNOU Books
❖ Software Engineering: IGNOU Books
❖ MCSD Preparations Guide Exam 305-316
❖ RDBMS: IGNOU Books.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.

Note: The names we used in this it is a dummy name and its not related to any company or college or any person.