Fall 2015

**E-Classroom for an Underserved Institution**

Bhanuprakash Madupati  
*Governors State University*

Kaleem Danish Mohammed  
*Governors State University*

Dilipkumar Pampana  
*Governors State University*

Follow this and additional works at: [http://opus.govst.edu/capstones](http://opus.govst.edu/capstones)

Part of the [Databases and Information Systems Commons](https://opus.govst.edu/capstones), [Graphics and Human Computer Interfaces Commons](https://opus.govst.edu/capstones), [Instructional Media Design Commons](https://opus.govst.edu/capstones), and the [Systems Architecture Commons](https://opus.govst.edu/capstones)

**Recommended Citation**

Madupati, Bhanuprakash; Mohammed, Kaleem Danish; and Pampana, Dilipkumar, "E-Classroom for an Underserved Institution" (2015). *All Capstone Projects*. 153.  
[http://opus.govst.edu/capstones/153](http://opus.govst.edu/capstones/153)

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to [http://www.govst.edu/Academics/Degree_Programs_and_Certifications/](http://www.govst.edu/Academics/Degree_Programs_and_Certifications/)

Visit the [Governors State Computer Science Department](http://www.govst.edu/Academics/Computer_Science/)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact [opus@govst.edu](mailto:opus@govst.edu).
Table of Contents

1. INTRODUCTION
   1.1 INTRODUCTION TO PROJECT ................................................................. 4
   1.2 ORGANIZATION PROFILE ................................................................. 4
   1.3 PURPOSE OF THE SYSTEM ............................................................... 6
   1.4 PROBLEMS IN EXISTING SYSTEM .................................................... 6
   1.5 SOLUTION OF THESE PROBLEMS ....................................................... 6

2. SYSTEM ANALYSIS
   2.1 INTRODUCTION .................................................................................. 8
   2.2 ANALYSIS MODEL ........................................................................... 8
   2.3 STUDY OF THE SYSTEM ................................................................. 11
   2.4 SYSTEM REQUIREMENT SPECIFICATIONS ......................................... 13
   2.5 PROPOSED SYSTEM ......................................................................... 14
   2.6 INPUT AND OUTPUT ......................................................................... 14
   2.7 PROCESS MODULES USED WITH JUSTIFICATION ............................... 15

3. FEASIBILITY REPORT
   3.1 TECHNICAL FEASIBILITY ................................................................. 17
   3.2 OPERATIONAL FEASIBILITY ............................................................ 18
   3.3 ECONOMICAL FEASIBILITY ............................................................ 18

4. SOFTWARE REQUIREMENT SPECIFICATIONS
   4.1 FUNCTIONAL REQUIREMENTS ......................................................... 19
   4.2 PERFORMANCE REQUIREMENTS ................................................... 24

5. SELECTED SOFTWARE
   5.1 INTRODUCTION TO .NET FRAME WORK ........................................... 25
   5.2 ASP.NET ......................................................................................... 27
E-Class Room

5.3 C#.NET .................................................................27
5.4 SQL SERVER .................................................................29

6. SYSTEM DESIGN
6.1 INTRODUCTION ...............................................................31
6.2 NORMALIZATION .............................................................31
6.3 E-R DIAGRAM .................................................................32
6.4 DATA FLOW DIAGRAMS ................................................34
6.5 DATA DICTIONARY ............................................................43
6.6 UML DIAGRAMS ...............................................................48

7. OUTPUT SCREENS...............................................................59

8. SYSTEM TESTING AND IMPLEMENTATION
8.1 INTRODUCTION ...............................................................65
8.2 STRATEGIC APPROACH OF SOFTWARE TESTING ..................65

9. SYSTEM SECURITY
9.1 INTRODUCTION ...............................................................67
9.2 SECURITY IN SOFTWARE ................................................67

10. CONCLUSION .................................................................69

11. BIBLIOGRAPHY ..................................................................70
1. INTRODUCTION

1.1 INTRODUCTION TO PROJECT

The E-Class Room system is a web based project. An educational institution in India is understaffed and has limited interaction among faculty, student and industry experts. The project is to provide an online platform for the students and faculty of the institution to enhance their educational needs and to share their learning with their fellow students, faculty or industrial experts. It aims to provide a platform for mutual cooperation between different kinds of learning. The new system will provide directional way for online learning between faculty, student and industrial experts.

1.2 ORGANIZATION PROFILE

Software Solutions is an IT solution provider for a dynamic environment where business and technology strategies converge. Their approach focuses on new ways of business combining IT innovation and adoption while also leveraging an organization’s current IT assets. Their work with large global corporations and new products or services and to implement prudent business and technology strategies in today’s environment.

RANGE OF EXPERTISE INCLUDES:

- Software Development Services
- Engineering Services
- Systems Integration
- Customer Relationship Management
- Product Development
- Electronic Commerce
- Consulting
- IT Outsourcing

We apply technology with innovation and responsibility to achieve two broad objectives:

- Effectively address the business issues our customers face today.
- Generate new opportunities that will help them stay ahead in the future.
THIS APPROACH RESTS ON:

- A strategy where we architect, integrate and manage technology services and solutions - we call it AIM for success.
- A robust offshore development methodology and reduced demand on customer resources.
- A focus on the use of reusable frameworks to provide cost and times benefits.

They combine the best people, processes and technology to achieve excellent results consistency. We offer customers the advantages of:

SPEED:

They understand the importance of timing, of getting there before the competition. A rich portfolio of reusable, modular frameworks helps jump-start projects. Tried and tested methodology ensures that we follow a predictable, low-risk path to achieve results. Our track record is testimony to complex projects delivered within and even before schedule.

EXPERTISE:

Our teams combine cutting edge technology skills with rich domain expertise. What’s equally important - they share a strong customer orientation that means they actually start by listening to the customer. They’re focused on coming up with solutions that serve customer requirements today and anticipate future needs.

A FULL SERVICE PORTFOLIO:

They offer customers the advantage of being able to Architect, integrate and manage technology services. This means that they can rely on one, fully accountable source instead of trying to integrate disparate multi vendor solutions.

SERVICES:

providing its services to companies which are in the field of production, quality control etc with their rich expertise and experience and information technology they are in best position to provide software solutions to distinct business requirements.
1.3 **PURPOSE OF THE PROJECT**

The project is fully integrated with Customer Relationship Management (CRM) solution and developed in a manner that is easily manageable, time saving and relieving one from semi automated.

Developing a E-classroom system to promote a greater count of students to splurge into the field of Education. It integrates the benefits of a physical classroom with the convenience of a ‘no-physical-bar’ virtual learning environment, minus the commuting hazards and expenses. It will usher in the immense flexibility and sophistication in the existing learning platform structures, with the perfect blend of synchronous and asynchronous interaction. It provides a means of collaborative learning for the students.

The educational methodology utilized for the concept of the E-Classroom (a classroom in an electronic space) reflects asynchronous group communications and collaborative approaches to education and training. The student is an active part of a learning group but proceeds to learn and understand on an individual basis independent of the speed of other learners in the group.

The E-Classroom is a teaching and learning environment located within a computer-mediated communication system. The objectives of a E-Classroom are to improve access to advanced educational experiences by allowing students and instructors to participate in remote learning communities using personal computers at home or at work; and to improve the quality and effectiveness of education by using the computer to support a collaborative learning process. By collaborative learning is meant a learning process that emphasizes group or cooperative efforts among faculty and students, active participation and interaction on the part of both students and instructors, and new knowledge that emerges from an active dialog among those who are sharing ideas and information.

1.4 **PROBLEM IN EXISTING SYSTEM**

In this existing system, the user has to attend the classes in regular time only, once delivered course cannot be repeated. Flexibility of multiple courses learning is not possible in the system. And also the user has to copy important notes. If any exams are conducting he has to attend the exams. The course should be finished in time. Maintaining the data relating students and course not in secured manner. This system maintains the data in the form of excel sheets and devices.

1.5 **SOLUTION OF THESE PROBLEMS**

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach.

- User Friendliness is provided in the application with various controls provided by system Rich User Interface.
• It can be accessed over the Internet.
• The user information files can be stored in centralized database which can be maintained by the system.
• This can give the good security for user information because data is not in client machine.
• Authentication is provided for this application only registered members can access.

As the E-classroom is one that aims to give the student an experience equal to or better than the sort they would find in a traditional classroom, there are obviously many advantages of the E-classroom to the student, as well as the teacher (or instructor) and the associated educational institution.
2. SYSTEM ANALYSIS

2.1 INTRODUCTION

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

2.2 ANALYSIS MODEL

SDLC METHODOLOGIES

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning and designing the second prototype.
4. Constructing and testing the second prototype.

At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.

The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

The final system is constructed, based on the refined prototype.

The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.
The following diagram shows how a spiral model acts like:

1. Determine objectives
2. Identify and resolve risks
3. Development and Test
4. Plan the next iteration

Fig 1.0-Spiral Model
2.3 STUDY OF THE SYSTEM

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI’S at the top level have been categorized as

1. Administrative user interface

2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Data updating along with the extensive data search capabilities.

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities

NUMBER OF MODULES

The system after careful analysis has been identified to be presented with the following modules:

- Administrators /Managers
- Faculties
- Students
- Web Registration
- Virtual Training
- Exam Conducting
- Search
- Reports
- Authentication
**Web Registration**

The system has a process of registration. Every student needs to submit his complete details in the form of registration. Whenever a student registration is completed automatically student can get a user id and password. By using that user id and password student can log into the system. If registration process is completed successfully an email remainder is generated and sends to the citizen.

**Administrators**

Administrator is treated as a super user in this system. He can have all the privileges to do anything in this system. Admin can prepare course curriculum, along with course content by the help of the faculties. He can collect the course videos and stored into the centralized database.

Another task done by the administrator is he can generates reports, log files, backup, and recovery of data any time.

**Faculty**

Faculties can take lectures, uploads assignments, announcements, evaluate answer sheets and also upload lectures, and other discussions in various formats as in the form of videos and power point presentations.

**Students**

Students can choose courses, attend lectures, take exams, and view their attendance records, progress reports as per their convenience through this system only.

One student can register multiple courses. Take their classes on the schedule time or later their convenience. By using the system interface student learn the courses properly and attend the exams as an online mode. Results will be announced through online only.

**Virtual Training**

Asynchronous communication in the form of emails, discussion boards that enable communication to occur at “convenient-times” that suit student schedules and are not accessed at simultaneous or prearranged times.
Exam Conducting

When ever a student is completed his course, an exam should be conducted, students need to attend the exam properly. It’s an online exam. The results will be announced later after the papers are evaluated by the faculties.

Reports

Different kind of reports is generated by the system.

- Student Progress report Request.
- Faculty performance report.
- Time based and request attendance reports for the students.

Authentication

Authentication is the process the where it provides security for application.

The system checks credentials whether the user is valid or not. If the user is valid system allows for authorization. Otherwise system will throw back and gives message

2.4 System Requirement Specifications

Hardware Requirements:

Processor: Intel Pentium or More

RAM: 1GB Ram

Hard Disk: PC with 80GB

Software Requirements:

- WINDOWS OS (XP / 2000 / 200 Server / 2003 Server)
- Internet Information Server 5.0 (IIS)
- Visual Studio .Net Framework (Minimal for Deployment) version 3.5
- SQL Server 2005 Enterprise Edition
2.5 PROPOSED SYSTEM

To debug the existing system, remove procedures those cause data redundancy, make navigational sequence proper. To provide information about users on different level and also to reflect the current work status depending on organization. To build strong password mechanism.

NEED FOR COMPUTERIZATION

We all know the importance of computerization. The world is moving ahead at lightning speed and everyone is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

- Minimizing the manual records kept at different locations.
- There will be more data integrity.
- Facilitating desired information display, very quickly, by retrieving information from users.
- Facilitating various statistical information which helps in decision-making?
- To reduce manual efforts in activities that involved repetitive work.

Updating and deletion of such a huge amount of data will become easier.

FUNCTIONAL FEATURES OF THE MODEL

As far as the project is developed the functionality is simple, the objective of the proposal is to strengthen the functioning of Audit Status Monitoring and make them effective and better. The entire scope has been classified into five streams knows as Coordinator Level, management Level, Auditor Level, User Level and State Web Coordinator Level. The proposed software will cover the information needs with respect to each request of the user group viz. accepting the request, providing vulnerability document report and the current status of the audit.

2.6 INPUT AND OUTPUT

The major inputs and outputs and major functions of the system are follows:

Inputs:

- Admin enter his user id and password for login.
• User enters his user id and password for login.
• User creates a new folder for personnel usage.
• Admin enters user id or date for tracking user login information.
• New user provides his completed personnel, address, and phone details for registration.
• Admin provides different user information for searching user data.
• User provides his user id, hint question, answer for getting the forgotten password.
• Administrator provides information to generate various kinds of reports.

Outputs:
• Admin can have his own home page.
• Users enter their own home page.
• The user defined folders can store in the centralized database.
• Admin will get the login information of a particular user.
• The new user’s data will be stored in the centralized database.
• Admin gets the search details of different criteria.
• User can get his forgotten password.
• Different kinds of reports are generated by the administrator.

2.7 PROCESS MODEL USED WITH JUSTIFICATION

Access control for data which require user authentication
The following commands specify access control identifiers and they are typically used to authorize and authenticate the user (command codes are shown in parentheses)

USER NAME (USER)

The user identification is that which is required by the server for access to its file system. This command will normally be the first command transmitted by the user after the control connections are made (some servers may require this).
PASSWORD (PASS)

This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control.
3. FEASIBILITY REPORT

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

3.1. TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source.
The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.2. OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

3.3. ECONOMICAL FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economical feasibility for certain.
4. SOFTWARE REQUIREMENT SPECIFICATION

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope: This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

DEVELOPERS RESPONSIBILITIES OVERVIEW:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

4.1. FUNCTIONAL REQUIREMENTS

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization.
- Internal Outputs whose destination is within organization and they are the User’s main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.
OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

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

Output Media:

In the next stage it is to be decided that which medium is the most appropriate for the output. The main considerations when deciding about the output media are:

- The suitability for the device to the particular application.
- The need for a hard copy.
- The response time required.
- The location of the users
- The software and hardware available.

Keeping in view the above description the project is to have outputs mainly coming under the category of internal outputs. The main outputs desired according to the requirement specification are:

The outputs were needed to be generated as a hot copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.
INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department’s communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;
• Type of input
• Flexibility of format
• Speed
• Accuracy
• Verification methods
• Rejection rates
• Ease of correction
• Storage and handling requirements
• Security
• Easy to use
• Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As input data is to be directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**ERROR AVOIDANCE**

At this stage, care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

**ERROR DETECTION**

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

**DATA VALIDATION**

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in,
the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

**USER INTERFACE DESIGN**

It is essential to consult the system users and discuss their needs while designing the user interface:

**USER INTERFACE SYSTEMS CAN BE BROADLY CLASSIFIED AS:**

1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
2. Computer initiated interfaces

   In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

**USER_INITIATED INTERFACES**

User initiated interfaces fall into tow approximate classes:

1. Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
2. Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms oriented interface is chosen because it is the best choice.

**COMPUTER-INITIATED INTERFACES**

The following computer – initiated interfaces were used:

1. The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
2. Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.
Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

**ERROR MESSAGE DESIGN:**

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed. This application must be able to produce output at different modules for different inputs.

**4.2. PERFORMANCE REQUIREMENTS**

Performance is measured in terms of the output provided by the application.

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

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.
5. SELECTED SOFTWARE

5.1 INTRODUCTION TO .NET FRAMEWORK

The Microsoft .NET Framework is a software technology that is available with several Microsoft Windows operating systems. It includes a large library of pre-coded solutions to common programming problems and a virtual machine that manages the execution of programs written specifically for the framework. The .NET Framework is a key Microsoft offering and is intended to be used by most new applications created for the Windows platform.

The pre-coded solutions that form the framework's Base Class Library cover a large range of programming needs in a number of areas, including user interface, data access, database connectivity, cryptography, web application development, numeric algorithms, and network communications. The class library is used by programmers, who combine it with their own code to produce applications.

Programs written for the .NET Framework execute in a software environment that manages the program's runtime requirements. Also part of the .NET Framework, this runtime environment is known as the Common Language Runtime (CLR). The CLR provides the appearance of an application virtual machine so that programmers need not consider the capabilities of the specific CPU that will execute the program. The CLR also provides other important services such as security, memory management, and exception handling. The class library and the CLR together compose the .NET Framework.

ARCHITECTURE
Class library

<table>
<thead>
<tr>
<th>Namespaces in the BCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
</tr>
<tr>
<td>System. CodeDom</td>
</tr>
<tr>
<td>System. Collections</td>
</tr>
<tr>
<td>System. Diagnostics</td>
</tr>
<tr>
<td>System. Globalization</td>
</tr>
<tr>
<td>System. IO</td>
</tr>
<tr>
<td>System. Resources</td>
</tr>
<tr>
<td>System. Text</td>
</tr>
<tr>
<td>System.Text.RegularExpressions</td>
</tr>
</tbody>
</table>

Microsoft .NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built-in APIs are part of either System.* or Microsoft.* namespaces. It encapsulates a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. The .NET class libraries are available to all .NET languages. The .NET Framework class library is divided into two parts: the Base Class Library and the Framework Class Library.
5.2 ASP.NET

SERVER APPLICATION DEVELOPMENT

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

5.3 C#.NET

ADO.NET OVERVIEW

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind.

ADO.NET uses some ADO objects, such as the Connection and Command objects, and also introduces new objects. Key new ADO.NET objects include the Dataset, Data Reader, and Data Adapter.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the Dataset -- that is separate and distinct from any data stores. Because of that, the Dataset functions as a standalone entity. You can think of the Dataset as an always disconnected recordset that knows nothing about the source or destination of the data it contains. Inside a Dataset, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

A DataAdapter is the object that connects to the database to fill the Dataset. Then, it connects back to the database to update the data there, based on operations performed while the Dataset held the data. In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around chunks of information. At the center of this approach is the DataAdapter, which provides a bridge to retrieve and save data between a Dataset and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.
The XML-based **DataSet** object provides a consistent programming model that works with all models of data storage: flat, relational, and hierarchical. It does this by having no 'knowledge' of the source of its data, and by representing the data that it holds as collections and data types. No matter what the source of the data within the **DataSet** is, it is manipulated through the same set of standard APIs exposed through the **DataSet** and its subordinate objects.

**DATAADAPTERS (OLEDB/SQL)**

The **DataAdapter** object works as a bridge between the **DataSet** and the source data. Using the provider-specific **SqlDataAdapter** (along with its associated **SqlCommand** and **SqlConnection**) can increase overall performance when working with a Microsoft SQL Server databases. For other OLE DB-supported databases, you would use the **OleDbDataAdapter** object and its associated **OleDbCommand** and **OleDbConnection** objects.

The **DataAdapter** object uses commands to update the data source after changes have been made to the **DataSet**. Using the **Fill** method of the **DataAdapter** calls the SELECT command; using the **Update** method calls the INSERT, UPDATE or DELETE command for each changed row. You can explicitly set these commands in order to control the statements used at runtime to resolve changes, including the use of stored procedures. For ad-hoc scenarios, a **CommandBuilder** object can generate these at run-time based upon a select statement. However, this run-time generation requires an extra round-trip to the server in order to gather required metadata, so explicitly providing the INSERT, UPDATE, and DELETE commands at design time will result in better run-time performance.

1. ADO.NET is the next evolution of ADO for the .Net Framework.
2. ADO.NET was created with n-Tier, statelessness and XML in the forefront. Two new objects, the **DataSet** and **DataAdapter**, are provided for these scenarios.
3. ADO.NET can be used to get data from a stream, or to store data in a cache for updates.
4. There is a lot more information about ADO.NET in the documentation.
5. Remember, you can execute a command directly against the database in order to do inserts, updates, and deletes. You don't need to first put data into a **DataSet** in order to insert, update, or delete it.

Also, you can use a **DataSet** to bind to the data, move through the data, and navigate data relationships.
A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

**SQL SERVER TABLES**

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

**PRIMARY KEY**

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

**RELATIONAL DATABASE**

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational
database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

**FOREIGN KEY**

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

**REFERENTIAL INTEGRITY**

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

**DATA ABSTRACTION**

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

**Physical level:** This is the lowest level of abstraction at which one describes how the data are actually stored.

**Conceptual Level:** At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.

**View level:** This is the highest level of abstraction at which one describes only part of the database.

**6.SYSTEM DESIGN**
6.1. INTRODUCTION

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities - design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

6.2 NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.
**Insertion anomaly**: Inability to add data to the database due to absence of other data.

**Deletion anomaly**: Unintended loss of data due to deletion of other data.

**Update anomaly**: Data inconsistency resulting from data redundancy and partial update

**Normal Forms**: These are the rules for structuring relations that eliminate anomalies.

**6.3 E-R Diagrams**

The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.

The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.

The set of primary components that are identified by the ERD are

- Data object
- Relationships
- Attributes
- Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.
6.4 DATA FLOW DIAGRAMS

A data flow diagram is a graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

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

1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The later is usually indicated however by two separate arrows since these happen at different type.

2) A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.

3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.

4) A Data flow to a data store means update (delete or change).

5) A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.
DFD Diagrams:

Context 0th Level Diagram:

- ADMIN
- FACULTY
- STUDENT

Data Storage
- Admin
- Faculty
- Student
- General Users

Managerial
- UI Screens
- Reports

Virtual Class Room System Process

Data Input Stage
Data Out Put Stage

Login DFD Diagram:

- Open Login form
- Enter User Name and Password
- Check User
- User Home Page

Yes
Yes
No

Verify Data
Admin Details Data Flow:

1st level DFD:

- Open Form()
- Enter Login Details
- Validates Data
- Manage Faculty
- Manage Student
- Manage Mails
- Manage Courses
- Generate Reports
- Log out
- Data Storage
2nd Level DFD

Manage Faculty
1.2.1

Displays Faculty Details
1.2.2

Accept Faculty Registration
1.2.3

Displays Faculty on Courses
1.2.4

Manage Student
1.3.1

Displays Student Details
1.3.2

Accept Student Registration
1.3.3

Displays Student on Course
1.3.4

Student Master

Faculty Master

Faculty Course Master

TblStdCourse
Faculty Details Data Flow

1st level DFD
Student Details Data Flow

1st level DFD

Open Form() 3.0.0

Enter Login Details 3.0.1

Login Master

Manage personal Profile 3.0.3

New User Sign Up 3.0.2

View Course Material 3.0.4

Assignments 3.0.6

Examinations 3.0.7

Assignments

Examinations

Login Master 3.0.0

Manage Mails 3.0.5

Course Master

Mails Master

Student Master

Verification

Validation

NOTE: The diagram illustrates the flow of data and processes within a Student Details system. The arrows indicate the direction of data flow from one process to another.
2nd Level DFD

- Manage Mails
  - 3.0.5.1
- Compose Mail
  - 3.0.5.2
- Inbox
  - 3.0.5.3
- Address Book
  - 3.0.5.5
- OutBox
  - 3.0.5.4

- Verifies Data
- Insert
- Mails Master
### 6.5 DATA DICTIONARY

After carefully understanding the requirements of the client the entire data storage requirements are divided into tables. The below tables are normalized to avoid any anomalies during the course of data entry.

#### Tables Design:

**tbl_AssignmentRemarks**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignMentId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>RemarksToStudent</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
</tbody>
</table>

**tbl_ChatRoomMaster**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChatRoomId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>ChatRoomName</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>CourseId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
</tbody>
</table>

**tbl_CourseMaster**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>CName</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>CourseDescription</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>CourseDuration</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>CourseContentsUpload</td>
<td>varbinary(MAX)</td>
<td>✔</td>
</tr>
<tr>
<td>NoOfLectures</td>
<td>int</td>
<td>✔</td>
</tr>
</tbody>
</table>

**tbl_EmailDetails**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmailId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FromStudentFacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>ToStudentFacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FromStudentFaculty</td>
<td>bit</td>
<td>✔</td>
</tr>
<tr>
<td>ToStudentFaculty</td>
<td>bit</td>
<td>✔</td>
</tr>
</tbody>
</table>
### tbl_CourseFacultiesMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CourseId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>DescriptionByfaculty</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
</tbody>
</table>

### tbl_ExaminationSchedule

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExaminationId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>ExaminationDate</td>
<td>datetime</td>
<td>✔</td>
</tr>
<tr>
<td>ExaminationEndDate</td>
<td>datetime</td>
<td>✔</td>
</tr>
<tr>
<td>CourseID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>LectureID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>PassPercentageDecided</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
</tbody>
</table>

### tbl_FacultyMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>FacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyFirstName</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyMiddleName</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyLastName</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>datetime</td>
<td>✔</td>
</tr>
<tr>
<td>DateOfRegistration</td>
<td>datetime</td>
<td>✔</td>
</tr>
<tr>
<td>Address</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Email</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Phone</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Image</td>
<td>varbinary(MAX)</td>
<td>✔</td>
</tr>
<tr>
<td>SpecialRemarks</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Experience</td>
<td>datetime</td>
<td>✔</td>
</tr>
</tbody>
</table>

### tbl_FacultiesOnlineMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnlineID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>OnlineDateOfBirth</td>
<td>datatetime</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyId</td>
<td>int</td>
<td>✔</td>
</tr>
</tbody>
</table>
### tbl_ExamQuestionMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuestionID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>QuestionText</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Answers1</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Answers2</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Answers3</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Answers4</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>Answers5</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>CorrectAnswer</td>
<td>varchar(30)</td>
<td>✔</td>
</tr>
<tr>
<td>ExaminationId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>Marks</td>
<td>int</td>
<td>✔</td>
</tr>
</tbody>
</table>

### tbl_NewsLetterMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>NewsletterId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>LetterUploadFile</td>
<td>varbinary(MAX)</td>
<td>✔</td>
</tr>
<tr>
<td>Date</td>
<td>datetime</td>
<td>✔</td>
</tr>
<tr>
<td>FacultyID</td>
<td>int</td>
<td>✔</td>
</tr>
</tbody>
</table>

### tbl_StudentAssignmentSubmissionMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssignmentId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>StudentID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>SubmissionDate</td>
<td>datetime</td>
<td>✔</td>
</tr>
<tr>
<td>LectureID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>SubmissionFile</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
</tbody>
</table>

### tbl_StudentExaminationDetails

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentExamId</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>QuestionID</td>
<td>int</td>
<td>✔</td>
</tr>
<tr>
<td>Answer</td>
<td>varchar(50)</td>
<td>✔</td>
</tr>
<tr>
<td>CorrectWrongbit</td>
<td>bit</td>
<td>✔</td>
</tr>
</tbody>
</table>
## tbl_StudentMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>StudentFirstName</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>StudentMiddleName</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>StudentLastName</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>datetime</td>
<td>☑</td>
</tr>
<tr>
<td>DateOfRegistration</td>
<td>datetime</td>
<td>☑</td>
</tr>
<tr>
<td>Address</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>Email</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>Phone</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>Image</td>
<td>varbinary(MAX)</td>
<td></td>
</tr>
</tbody>
</table>

## tbl_StudentCourseMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>CourseId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>FacultyId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>CourseRegistrationId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>RegistrationId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>RegistrationDate</td>
<td>datetime</td>
<td>☑</td>
</tr>
<tr>
<td>CourseEndDate</td>
<td>datetime</td>
<td>☑</td>
</tr>
</tbody>
</table>

## tbl_UserMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>UserName</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>Password</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
<tr>
<td>StudentOrFacultyId</td>
<td>int</td>
<td>☑</td>
</tr>
<tr>
<td>StudentOrFacultyBit</td>
<td>bit</td>
<td>☑</td>
</tr>
<tr>
<td>UserActivateState</td>
<td>varchar(50)</td>
<td>☑</td>
</tr>
</tbody>
</table>
### tbl_StudentAttendanceMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttendanceID</td>
<td>int</td>
<td>no</td>
</tr>
<tr>
<td>AttendanceDateTime</td>
<td>datetime</td>
<td>yes</td>
</tr>
<tr>
<td>DisconnectedDateTime</td>
<td>datetime</td>
<td>yes</td>
</tr>
<tr>
<td>StudentID</td>
<td>int</td>
<td>yes</td>
</tr>
<tr>
<td>CourseID</td>
<td>int</td>
<td>yes</td>
</tr>
<tr>
<td>LectureID</td>
<td>int</td>
<td>yes</td>
</tr>
</tbody>
</table>

### tbl_VoiceMailBoxMaster

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoiceMailBoxId</td>
<td>int</td>
<td>no</td>
</tr>
<tr>
<td>VoiceMailBoxText</td>
<td>varchar(30)</td>
<td>yes</td>
</tr>
<tr>
<td>StudentID</td>
<td>int</td>
<td>yes</td>
</tr>
<tr>
<td>Audiofile</td>
<td>varbinary(MAX)</td>
<td>yes</td>
</tr>
</tbody>
</table>
6.6 UML DIAGRAMS

Use Case Diagrams :

Over View Use Case Diagram :

![Over View Use Case Diagram](image-url)
Admin Use Diagram:

Faculty Use Case Diagram:
Activity Diagrams:

Registration Diagram:

[Enter Registration Details]

- Get the Details
- [submit]
- Validate Details

- No → Rejected
- Yes → Accepted

[Success Fully Registered]
Login Activity Diagram:

[Enter User Name and Password]

Get Details

[Submit]

Validate Data

Rejected

No

Accepted

yes

Error
Admin Activity Diagram:

[Enter User Name and Password]

Get the Details

[Submit]

Validate Data

NO

[Generate Reports] [Provide Services]

Get the Details

Get the Details

YES

NO

[Successfully Done]
Faculty Activity Diagram:

1. [Enter User Name and Password]
2. Get the Details
3. [Submit]
4. Validate Data
   - NO
   - Yes
5. [Upload Files]
6. [View the Queries]
   - Validate Data
     - NO
     - Yes
   - Validate Data
     - NO
   - Yes
7. [Successfully Done]
Student Activity Diagram:

[Enter User Name and Password] → Get the Details → [Submit] → Validate Data

Validate Data: NO → Yes

[View Personal Details] → Validate Data

[Download Course Files] → Validate Data

NO → NO

Yes → Yes → [Successfully Done]
Admin Login Sequence Diagram

Sequence Diagram For Faculty Acceptance

Sequence Diagram for Student Login
Adding new course by admin

Student Course registration
1: GetRegistration()

2: :: insertFacultyData()

3: ExecuteNonQuery()

4: ExecuteCommand()

5: Response()

6: Response()

7: Response()
7. OUTPUT SCREENS

Student and Professor Sign Up page

Admin, Student, Faculty Login Page
List of Faculty Page

Adding new course
Managing existing Course

Faculty Course and Achievement Details
Assigning Your Assignment For Student

Student Assignment details
Student Submitting Assignment page

Remarks for Student assignment
Chat Room Between Faculty and student
8. SYSTEM TESTING AND IMPLEMENTATION

8.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

8.2. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.
9. SYSTEM SECURITY

9.1 INTRODUCTION

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

System Security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

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

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

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

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

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

9.3 SECURITY SOFTWARE

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employees two types of checks and controls:
CLIENT SIDE VALIDATION

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

- VBScript in used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.
- Tab-indexes are set according to the need and taking into account the ease of user while working with the system.

SERVER SIDE VALIDATION

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

- Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.

- User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.

- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled o the server side.
10. CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in ASP.NET and C#.NET web based application and no some extent Windows Application and SQL Server, but also about all handling procedure related with “E-Class Room”. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows:

- It’s a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can sat that the project is user friendly which is one of the primary concerns of any good project.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.

LIMITATIONS:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.
# BIBLIOGRAPHY

<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
<th>Author</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASP.NET Projects - Building 10 Enterprises Projects (with CD)</td>
<td>Eric A. Smith</td>
<td>2002</td>
</tr>
<tr>
<td>2</td>
<td>ASP.NET - The Complete Reference</td>
<td>Matthew MacDonald</td>
<td>Latest</td>
</tr>
<tr>
<td>3</td>
<td>ASP.NET 3.5 : A Beginner's Guide</td>
<td>William B. Sanders</td>
<td>2009</td>
</tr>
<tr>
<td>4</td>
<td>ASP.NET 3.5 AJAX Unleashed</td>
<td>Robert Foster</td>
<td>2009</td>
</tr>
<tr>
<td>5</td>
<td>ASP.NET 3.5:COVERS C# &amp; VB 2008 CODES, BLACK BOOK, PLATINUM ED (BLACK BOOK series) (With CD VCD DVD)</td>
<td>KOGENT SOLUTIONS INC.</td>
<td>2009</td>
</tr>
<tr>
<td>6</td>
<td>ASP.NET 3.5 IN SIMPLE STEPS (SIMPLE STEPS series)</td>
<td>KOGENT SOLUTIONS INC.</td>
<td>2009</td>
</tr>
<tr>
<td>7</td>
<td>ASP.NET Unleashed (Book + CD-ROM)</td>
<td>Stephen Walther</td>
<td>2009</td>
</tr>
<tr>
<td>8</td>
<td>ASP.NET 4.0: COVERS C# 2010 AND VB 2010 CODES, BLACK BOOK (BLACK BOOK series) (With CD VCD DVD)</td>
<td>KOGENT LEARNING SOLUTIONS INC.</td>
<td>2010</td>
</tr>
<tr>
<td>9</td>
<td>ASP.NET 4.0 PROJECTS: COVERS .NET 3.5 AND .NET 4.0 CODES, BLACK BOOK (BLACK BOOK series) (With CD VCD DVD)</td>
<td>KOGENT LEARNING SOLUTIONS INC.</td>
<td>2010</td>
</tr>
<tr>
<td>10</td>
<td>ASP.NET 4.0 IN PRACTICE (MANNING series)</td>
<td>DANIELE BOCHICCHIO,STEFANO MOSTARDA,MARCO DE SANCTIS</td>
<td>2011</td>
</tr>
<tr>
<td>11</td>
<td>ASP.NET MVC IN ACTION (MANNING series)</td>
<td>JEFFREY PALERMO, BEN SCHEIRMAN, JIMMY BOGARD</td>
<td>2009</td>
</tr>
<tr>
<td>12</td>
<td>ASP.NET WITH C# (MISL-DT series)</td>
<td>KOGENT LEARNING SOLUTIONS INC.</td>
<td>2012</td>
</tr>
</tbody>
</table>