**1.SYNOPSIS**

“Lin\_WIN” is a virtual classroom. The objective of “Lin\_WIN” is to bring everything possible in a real-world classroom to the virtual world. This is an online learning point where the main objective is to teach programing languages. The basic technique used in this system is content sharing and management. In this system, everything required to make a programing language understandable to a student are provided like; Video lectures, word documents, ppts, pictures, pdf and more over a faculty is allocated to a course who will guide the students to complete the course. It is designed to make the maximum interaction among users so they can share their doubts and ideas.

**2.INTRODUCTION**

**2.1 SELECTION OF THE SOFTWARE**

Selection is an important part in designing the system. One reason is that the number of software in the market is experiencing a geometric progression. To choose the best from many alternatives as proved to be quite very ticklish problem. Software selection is the critical aspect of system development. The software must be reliable, usable. Secure and serviceable.

ABOUT ASP.Net

ASP.NET is a technology that allows us to build and control dynamic Web pages easily. It also provides many enhancements to take advantage of new technology as we can interact with databases, personalize Web pages for visitors, display page on mobile devices (such as cell phones), and even build an entire e-commerce site from scratch.

Previously internet works on *request/response* model that is an integral part of *client/server* model. Although this is a marvelous way of communicate and distribute information, it's rather simple and static. When the process is over, once client receives the requesting page from the server the server has no idea what the client is doing unless it makes another request.

There is another model for communicating between server and clients, known as *event-driven* model.ASP.NET work on this model, it detects action and responds to them i.e. the server waits around for something to happen on the client. Once it does, the server takes action and performs some piece of functionality. Of course, a Web, server cannot know what you are thinking, but it can respond to your actions. If you type some text on Web page, the server responds to it. If you click an image, the server responds.

SQL SERVER: ­

SQL Server is an SQL-compliant RDBMS. SQL-compliant means it use the ANSI (American National Standard Institute) version of Structured Query Language or ‘SQL’. Structured Query Language is a command that allow us to modify or retrieve information from the database. Client server means that SQL Server is designed to store data in the central location (the server) and deliver it on demand to numerous other locations (the client). SQL Server is also a Relational Database Management System (RDBMS).

SQL SERVER 2014 Features: ­

### 1. In-Memory OLTP Engine

SQL Server 2014 enables memory optimization of selected tables and stored procedures. The In-Memory OLTP engine is designed for high concurrency and uses a new optimistic concurrency control mechanism to eliminate locking delays. Microsoft states that customers can expect performance to be up to 20 times better than with SQL Server 2012 when using this new feature. For more information, check out “[Rev Up Application Performance with the In-Memory OLTP Engine](http://sqlmag.com/sql-server-2014/application-performance-inmemory-oltp-database-engine).”

### 2. Always On Enhancements

Microsoft has enhanced Always On integration by expanding the maximum number of secondary replicas from four to eight. Readable secondary replicas are now also available for read workloads, even when the primary replica is unavailable. In addition, SQL Server 2014 provides the new Add Azure Replica Wizard, which helps you create asynchronous secondary replicas in Windows Azure.

### 3. Buffer Pool Extension

SQL Server 2014 provides a new solid state disk (SSD) integration capability that lets you use SSDs to expand the SQL Server 2014 Buffer Pool as nonvolatile RAM (NvRAM). With the new Buffer Pool Extensions feature, you can use SSD drives to expand the buffer pool in systems that have maxed out their memory. Buffer Pool Extensions can provide performance gains for read-heavy OLTP workloads.

### 4. Updateable Column store Indexes

When Microsoft introduced the column store index in SQL Server 2012, it provided improved performance for data warehousing queries. For some queries, the column store indexes provided a tenfold performance improvement. However, to utilize the column store index, the underlying table had to be read-only. SQL Server 2014 eliminates this restriction with the new updateable Column store Index. The SQL Server 2014 Column store Index must use all the columns in the table and can’t be combined with other indexes.

### 5. Storage I/O control

The Resource Governor lets you limit the amount of CPU and memory that a given workload can consume. SQL Server 2014 extends the reach of the Resource Governor to manage storage I/O usage as well. The SQL Server 2014 Resource Governor can limit the physical I/Os issued for user threads in a given resource pool.

### 6. Power View for Multidimensional Models

Power View used to be limited to tabular data. However, with SQL Server 2014, Power View can now be used with multidimensional models (OLAP cubes) and can create a variety of data visualizations including tables, matrices, bubble charts, and geographical maps. Power View multidimensional models also support queries using Data Analysis Expressions (DAX).

### 7. Power BI for [Office 365](http://winsupersite.com/office-365) Integration

Power BI for Office 365 is a cloud-based business intelligence (BI) solution that provides data navigation and visualization capabilities. Power BI for Office 365 includes Power Query (formerly code-named Data Explorer), Power Map (formerly code-named GeoFlow), Power Pivot, and Power View. You can learn more about Power BI at [Microsoft’s Power BI for Office 365 site](http://www.microsoft.com/en-us/powerbi/default.aspx#fbid=uCAOUna9BrP).

### 8. SQL Server Data Tools for Business Intelligence

The new SQL Server Data Tools for BI (SSDT-BI) is used to create SQL Server Analysis Services (SSAS) models, SSRS reports, and SSIS packages. The new SSDT-BI supports SSAS and SSRS for SQL Server 2014 and earlier, but SSIS projects are limited to SQL Server 2014. In the pre-release version of SQL Server 2014, SQL Server Setup doesn’t install SSDT-BI. Instead, you must [download SSDT-BI separately](http://www.microsoft.com/en-us/download/details.aspx?id=36843) from the Microsoft Download Center.

### 9. Backup Encryption

One welcome addition to SQL Server 2014 is the ability to encrypt database backups for at-rest data protection. SQL Server 2014 supports several encryption algorithms, including Advanced Encryption Standard (AES) 128, AES 192, AES 256, and Triple DES. You must use a certificate or an asymmetric key to perform encryption for SQL Server 2014 backups.

### 10. SQL Server Managed Backup to Windows Azure

SQL Server 2014’s native backup supports Windows Azure integration. Although I’m not entirely convinced that I would want to depend on an Internet connection to restore my backups, on-premises SQL Server 2014 and Windows Azure virtual machine (VM) instances support backing up to Windows Azure storage. The Windows Azure backup integration is also fully built into SQL Server Management Studio (SSMS).

**2.2 ABOUT THE PROJECT**

“Lin\_WIN” is an online learning system. Aimed for teaching programing languages. “Lin\_WIN” have three level of actors or users. First one is administrator, administrator is the one who describe the courses and manages the faculties. Describing a course involves, assigning a faculty for the course, categorizing the whole course in different modules having number of topics; that is to set a syllabus for the course , then, scheduling exams, scheduling live chat with faculty, adding materials to the course..etc. Second one is the faculty, faculty is the one who manages the courses assigned to him/her as described by the admin. Faculty is responsible to provide proper material to the students and to make their queries clarified. Third one is students, students just need to register for a course and utilize the materials and faculty help at max and learn something new.

**2.4. SYSTEM REQUIREMENT SPECIFICATION**

**HARDWARE SPECIFICATION**

Processor - Pentium Dual Core or higher

RAM - 2 GB or higher

Hard Disk - 300 GB or higher

Keyboard - Entry Level Quiet Key Keyboard, PS/2

Mouse - Optical Mouse

Monitor - 15” Digital color / LCD /LED monitor

Display Type - VGA / SVGA

**SOFTWARE SPECIFICATION**

Operating System - Windows 7 or Higher

Application server - ISI

Scripts - JAVASCRIPT

Front End - ASP.Net C#

Database - SQL Server 2014

**3.SYSTEM ANALYSIS**

A system is an orderly grouping of interpedently components linked together according to a plan to achieve a specific objective. System Analysis is an activity that takes place when new information systems are being developed or existing system are being changed. The steps involved in system analysis are identifying a problem associated with a system, examining strengths and weakness of the old system and determining the form of new system. It also includes collection of necessary data and developing plans for the new systems.

**3.1. EXISTING SYSTEM**

The existing system is an ordinary classroom around four walls with lots of real world factors affecting. The existing system have to follow up some real world constraints like time, number of students in a class, outside world disturbance ..etc. The existing system follows a syllabus for a course and completes the course within a period of time. During the course period, faculties teach the students and conduct exams. On the basis of the exam conduct they allocate grades for the students.

**3.2. PROPOSED SYSTEM**

The demanded system is a virtual classroom which is capable of doing all the activities that are performed within a real-world classroom. So, the main objective was to understand the activities in a classroom and to develop an online application that behaves exactly like. The main advantage of this system is that a student can learn a new programing language by just being inside the home or where ever he/she wants from all around the world. Student can download the file, watch video, share their ideas and can clarify their doubts from the experts. No time limits, only other than that of the expert talk, where a student can make a live chat with the experts. A teacher can manage a course from being anywhere by just having a browser and an internet connection. So, the system is a better relaxation for both the students and teachers. No further special training is needed to operate this system because of its GUI anyone can handle it with ease, which is developed by considering the factor that anyone new to the system can understand the system easily.

The main modules in this project are;

* **Adding Courses:** Adding a course is the responsibility of the admin. While adding a new course the admin defines the following:
* **Syllabus Scheduling:** The complete course runs on the basis of a syllabus which is defined by the admin while adding the course. A syllabus includes number of modules and number of topics in each module.
* **Assigning Faculty:** A faculty is allocated by the admin to manage the course and clarify the student’s queries.
* **Examination Scheduling:** Every course has some exams in between the period of course. It is the responsibility of admin to schedule the exam.
* **Adding Materials:** Both admin and faculty can add materials to the course. Materials such as video lectures, different files; ppts, word documents, pdf.etc.
* **Managing the course:** Course management is the responsibility of a faculty where he adds new materials, responds to the queries by the students, conducts exams and assigns grade points on the basis of exam conducted.
* **Respond to course:**  Respond to course is a module where a student register for a course and utilizes the materials provided and make queries to faculty about the course.

**3.3. FEASIBILITY STUDY**

**Feasibility** is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

**3.3.1. TECHINICAL FEASIBILITY**

**Technical feasibility** assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements.

**3.3.2. SOCIAL FEASIBILITY**

**Social feasibility** assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed

**3.3.3. ECONOMIC FEASIBILITY**

**Economic feasibility** determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software.

**3.3.4. OPERATIONAL FEASIBILITY**

**Operational feasibility** refers to how well a planned system solves the problems of the project and how well it works at using the opportunities of the project. Operational feasibility is therefore one of the key element of feasibility.

**4.REQUIREMENT ANALYSIS**

Requirement analysis is a process of discovery refinement modeling and specification. The analysis can be divided into;

* Problem reorganization.
* Problem evaluation and synthesis.
* Modelling.

**4.1. PROBLEM RECOGNITION**

The goal of the step is recognition of basic problems of element as indicate by customer. The basic purpose of this activity is to obtain a thorough understanding of the needs of client and user what exactly is desired from the software is the constraints on the solution. The main problems of existing system are;

* Time consuming.
* More man power is wasted.

**4.2. PROBLEM EVALUATION AND SYNTHESIS**

In this step analyst must define all externally observable object, evaluating flow and control of step of information, define and elaborate all software function, understand the software behavior and design constraints etc. evaluation and synthesis continues until both analyst and customer felt confident about the project.

Once the problem identified, the evaluation process begins. After evaluation of the current problem and the desired information, the analyst may synthesis one or more solution.

* Cost effectiveness.
* Faster and records are accurate.
* More secure and easy retrieval of data.

**4.3. MODELING**

During a software requirement analysis, we create models to gain better understanding of actual logical entity to be built. The model helps the analyst to understanding information and function of the system. The model became the main reference for the review to determine completeness, the main method used for this is DFD.

**DATA FLOW DIAGRAM (DFD)**

A DFD , also known as the bubble chart , has the purpose of clarifying system requirements and identifying major information that will become programs in system design. A DFD is a pictorial representation of network that describes the flow of data through a system. The symbols used in Data Flow Diagram are;

* It represents a data source or destination.
* It represents flow of data.

* It represents a process that transforms data.
* It represents data storage (eg.table).

**5.SYSTEM DESIGN**

The system design specification includes input design specification and output design specification.

**5.1. ARCHITECTURAL DESIGN**

The architectural design defines the relationship between major structural elements of the system, and the constraints that affects the way in which the architectural design patterns can be applied. The architectural design that represents the framework of a computer based system and it can it can be derived from the system specification, the analysis model and the interaction of subsystem defined within the analysis model. Architectural design creates a blue print of the system design with necessary specification of hardware, software, people and data resources.

**5.2. PROCEDURAL DESIGN**

Procedural design or component level design data, architectural, and interface designs are must be translated into operational software. The procedural design for each component represents in graphical, tabular, or text notation and is the primary product produced during component level design.

**5.3. DATA DESIGN**

Data design creates a model of data or information that is represented at higher level of abstraction. The software of the data has always been an important part of the software design. The data design activity translates these elements of requirements model into data structure at the software component level.

**5.4. INTERFACE DESIGN**

Interface design creates an active communication medium between a human and computer design, identifiers, interfaces, objects and then creates a layout that forms the basis for a user interface.

**6.CODING**

A coding provides a brief identification of data items and replaces longer description that would be more awkward to store and manipulate. A code can be defined as a group of character used to identify an item of data while identification is a main function of a code. A code also shows relation between items of data.

The code plan identifies the particular characteristics that need to be contained within the code. Only information that makes possible efficient identification and retrieval of coded item should be chosen. The method chosen must have the following features.

* Expandable
* Precise
* Concise
* Meaningful
* Operable

**7.SYSTEM TESTING**

The testing phase is an important part of software development. It is the process of finding errors and missing operations and also complete verifications to determine whether the objectives and requirements are satisfied. Software testing is carried out in three steps;

The first step includes unit testing where in each module is tested to provide its correctness, to determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is important and major part of the project. So errors are rectified easily in particular modules and program quality is increased.

Second step include integration testing. If we need not be the case that software whose modules when run individually and showing perfect result as whole. These individual modules are clipped under a major module and tested again and verified the result.

The third step involves validation and testing which determines whether the software function as the user expected.

**7.1. TESTING PROCESS**

Testing is the process of executing the program with the intention of finding errors. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system elements have properly integrated and perform allocated functions.

**7.2. UNIT TESTING**

It is the process of taking each program modules and runs in isolation from the rest of the modules, by using prepared inputs and comparing the results with the results predicted by the specification and design of modules. This enables the tester to detect errors in coding and logic that are contained within the module.

**7.3. INTAGRATED TESTING**

Integration testing is a systematic technique for constructing tests to uncover errors associated within the interface. In this testing, all the modules are combined, and then the entire program is tested as whole.

**7.4. VALIDATION TESTING**

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and validation testing may begin. Validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

**8.SYSTEM IMPLEMENTATION**

System implementation is the process of bringing the developed system and turning it over to user. It can be the most crucial stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective.

**IMPLEMENTATION PLAN**

For the successful implementation of the system, implementation plan is necessary. Its major elements include test plan, training plans, an equipment installation plan and a conversion plan. The test plan provides for the preparations of the test ad for testing the system in a planned, structured manner. Training plan is necessary to ensure that all people who are associated with the computer related information system have the necessary knowledge and skills. He important activities are preparations, equipment installation and hardware- software checkout.

Conversion is the process if initiating and performing all the physical operations that result directly in the turnover of the new system to the user. There are two parts of conversion. The conversion plan implemented throughout the development phase into the operational phase. The conversion plan includes procedural conversions, programmed conversion and the file conversion. The changeover plan also specifies the method of change from old to new system. Choices of changeover methods include parallel operations, immediate replacement and physical change over.

**SECURITY TECHNOLOGIES AND POLICIES.**

The security features are considered while developing the system, so as to avoid the errors and omission that may lead to serious problems. That system may have to face the unwanted events called threats. Computer system is secured against a particular threat if counter measures have been taken to reduce an acceptability low level amount of loss that the threat may be expected to cause over a given period of time. A computer should be protected from the following three problems

1. Loss of availability.

2. Loss of integrity.

3. Loss of confidentiality.

These problems may adversely affect the ability of the computer system to carry out its intended task. Threat to a computer system is any event that adversely affects the one or more assets or resources, which make up the system.

In this project the data security, data validation checking methods are applied using a password authentication. All the data, which is entered by employee or administrator, will be validated. Users should not allow to login unless they enter their correct user id and password.

**9.SOFTWARE MAINTENENCE**

The maintenance is an important phase of any system. Maintenance of system should be done accurately and with specific care proper running of the system.

Maintenance involves the software industry captive, typing up the system resources. It means restoring something to its original conditions. Maintenance involves a wide range of activities including correcting, coding and design errors, updating documentation and test data, and upgrading user support. Maintenance is continued till the product is reengineered or deployed to another platform. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware or enhancing the software.

Any system developed should be secure and protected against possible hazards. the system should be maintained and upgraded according to the technological advancements. It ensures the data integrity; data control and security. The backup copies of data must be maintained daily so that to prevent the loss of data due to various reasons. Security measures are provided to prevent unauthorized access of the database at various levels. Software maintenance is divided into three categories;

* Corrective maintenance.
* Adaptive maintenance.
* Perfective maintenance.

Corrective maintenance has to do with the removal of residual errors present in the product when it is delivered as well as errors introduced into the software during its maintenance.

Adaptive maintenance modifies the software to keep it up date with its operative environment. It may be needed because of changes in user requirements.

Perfective maintenance involves changing the software to improve some of its qualities.

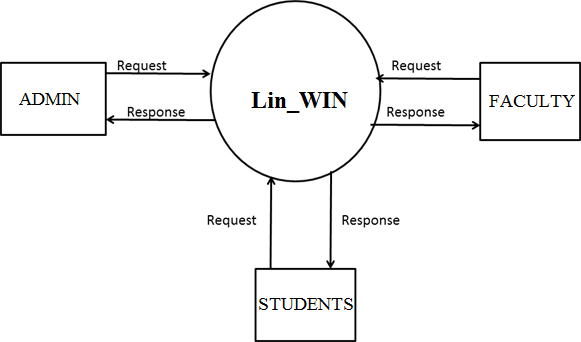
**10.CONCLUSION**

The project smart voting system was developed to solve the problems of manual system. The new system is able to reduce the work load of staffs. It can provide accuracy and security in data keeping; can reduce the man power. There is no need for keeping the written records the system will automatically store the records. Immediate result generation is another feature of the new system. By using this organization can handle election process in an easier way.

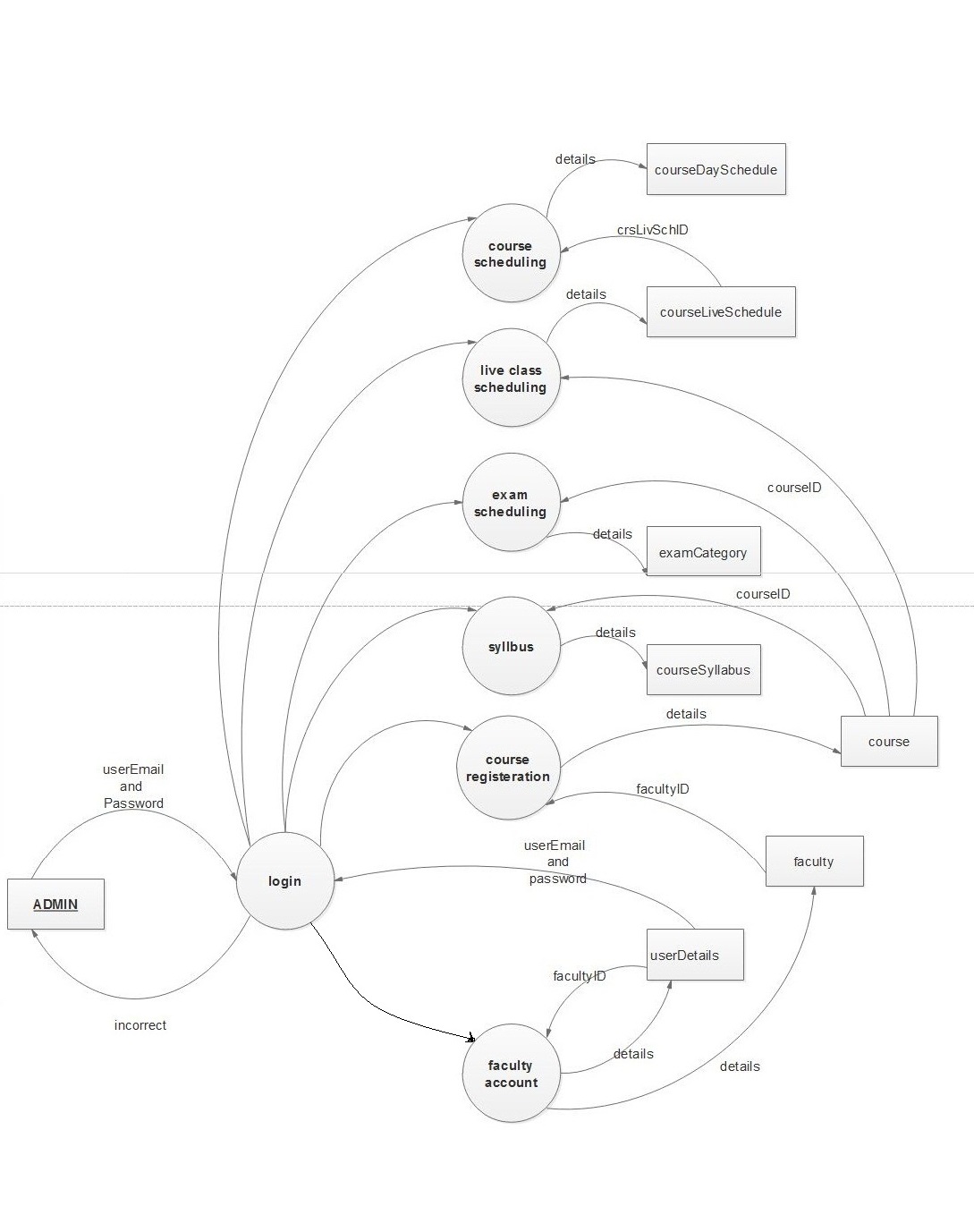
The new system is highly integrated and secured and has included validation. The new system is much faster than the old one. It is user friendly. One who has less computer knowledge can handle the system. As whole the new system has solved many problems of the old system. We hope the new system will be an advantage for all the end users.

**11.1-APPENDIX A(DATA FLOW DIAGRAM)**

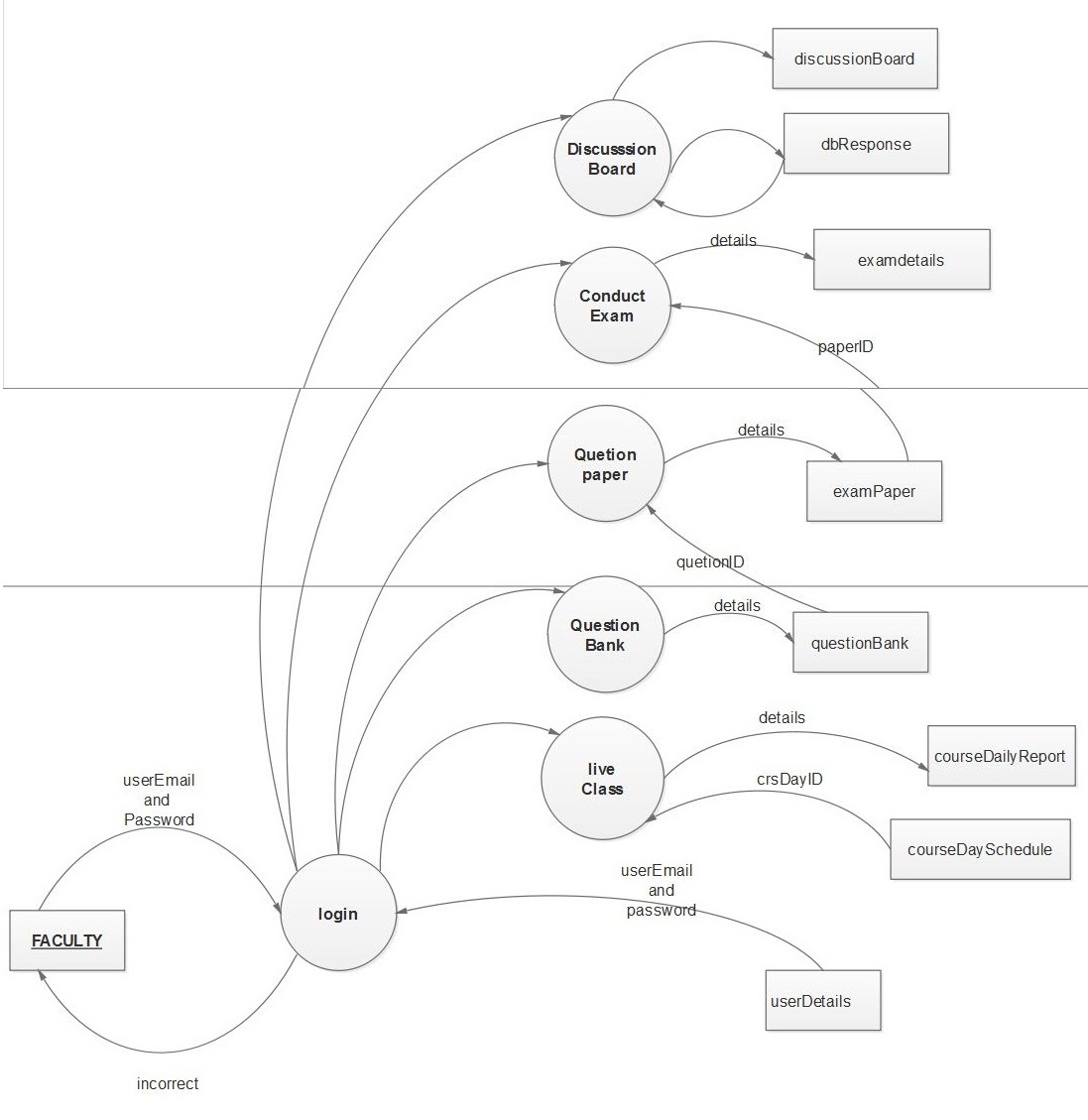
**CONTEXT LEVEL DIAGRAM**



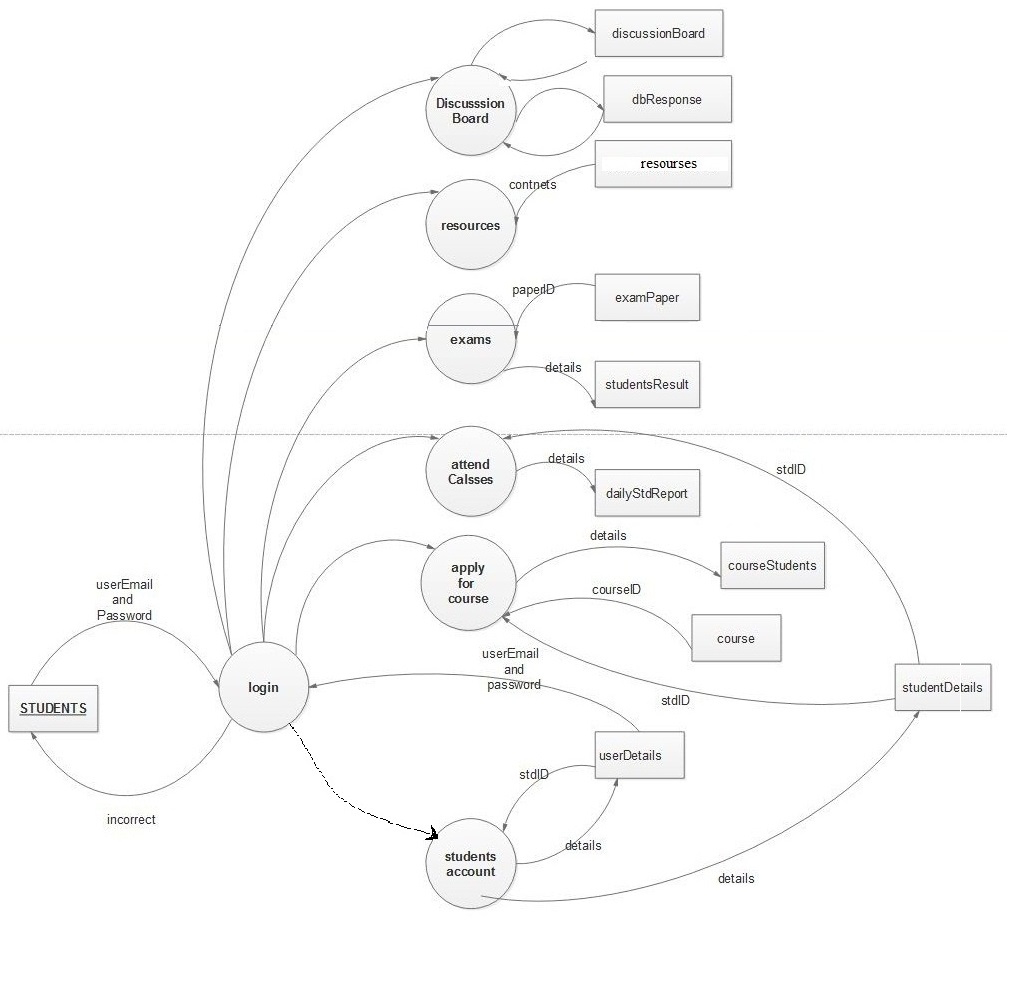
**LEVEL-1-FOR ADMIN**



**LEVEL-1- FOR RETURNING FACULTY**



**LEVEL-1- FOR STUDENTS**



**DATABASE DESIGN**

Once the input data entered in the system, these data may to be stored either for short or long period. The data will be stored in the database in a logical manner. To store data we use several tables;

**11.2-APPENDIX B(TABLES)**

**1. userDetails**

PRIMARY KEY: Email

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| userEmail | Varchar(50) | Email |
| Username | Varchar(50) | Name |
| userPassword | Varchar(50) | Password |
| profilePic | Varchar(50) | Profile Picture |
| userType | Varchar(50) | User’s Type |

FOREIGN KEY:

**2. studentDetails**

PRIMARY KEY: stdid

FOREIGN KEY: userEmail

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| stdId \* | Varchar(50) | Student’s Id |
| UserEmail# | Varchar(50) | Email |
| description | Varchar(500) | Description |
| dateOfJoin | DateTime | Date Of Join |

**3. faculty**

PRIMARY KEY: facultyID

FOREIGN KEY: userEmail

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| facultyID\* | Varchar(50) | Faculty Id |
| userEmail# | Varchar(50) | Email |
| description | Varchar(500) | Description |
| dateOfJoin | DateTime | Date Of Join |

**4. course**

PRIMARY KEY: courseID

FOREIGN KEY: facultyID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| coursed \* | Varchar(50) | Course ID |
| courseName | Varchar(50) | Course Name |
| description | Varchar(500) | Description |
| Image | Varchar(50) | Image |
| facultyID # | Varchar(50) | Faculty Id |
| startedOn | DateTime | Started On |
| Fees | Numeric(20) | Fees – Charge applied |
| courseDuration | Numeric(20) | Duration Of Course |
| Status | Varchar(50) | Running Status |

**5. courseSyllabus**

PRIMARY KEY: crsSyID

FOREIGN KEY: courseID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| crsSyID \* | Varchar(50) | Course Syllabus ID |
| courseID# | Varchar(50) | Course ID |
| topics | Varchar(50) | Topics |
| Modules | Varchar(50) | Modules |
| Deadline | Varchar(50) | Topics To be Covered till |

**6. courseBatch**

PRIMARY KEY: courseBatchID

FOREIGN KEY: courseID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| courseBatchID \* | Varchar(50) | Course Batch ID |
| startedON | DateTime | Batch Started On |
| CourseID# | Varchar(50) | Course ID |

**7.courseLiveSchedule**

PRIMARY KEY: crsLivSchID

FOREIGN KEY: courseBatchID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| crsLivSchID\* | Varchar(50) | Course Live Schedule ID |
| courseBatchID# | Varchar(50) | Course Batch ID |
| startsAt | Varchar(50) | Starting Time |
| Duration | Numeric(10) | Duration |

**8. courseDaySchedule**

PRIMARY KEY: crsDayID

FOREIGN KEY: crsLivSchID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DISCRIPTION** |
| crsDayID\* | Varchar(50) | Course Day ID |
| crsLivSchID# | Varchar(50) | Course Live Schedule ID |
| Day | Varchar(50) | Week Days |

**9. examCategory**

PRIMARY KEY: exmCatID

FOREIGN KEY: courseID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| exmCatID\* | Varchar(50) | Exam category ID |
| Name | Varchar(50) | Name Of Exam |
| maxMark | Numeric(20) | Maximum Marks |
| minMark | Numeric(20) | Minimum Marks |
| courseID# | Varchar(50) | Minimum Marks |
| noOfQuestions | Numeric(20) | Number Of Questions |
| moduleUpTo | Numeric(20) | Portions for Exam |

**10.materials**

PRIMARY KEY: materialID

FOREIGN KEY: crsSyID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| materialID\* | Varchar(50) | Extra fitting demanded id |
| Name | Varchar(50) | Extra fitting id |
| Description | Varchar(50) | Engine number |
| Class | Varchar(50) | Categorizing Materials |
| crsSyID# | Varchar(50) | Course Syllabus ID |
| Type | Varchar(50) | Type Of Material |
| lastUpdate | DAteTime | Last Update |
| Path | Varchar(50) | Directory Path |

**11. questionBank**

PRIMARY KEY: questionID

FOREIGN KEY: crsSyID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| questionID\* | Varchar(50) | Question ID |
| crsSyID# | Varchar(50) | Course Syllabus ID |
| question | Varchar(50) | Question |
| Answer | Varchar(50) | Answer |
| description | Varchar(500) | Description |
| opt1 | Varchar(50) | Option 1 |
| opt2 | Varchar(50) | Option 2 |
| opt3 | Varchar(50) | Option 3 |

**12. examDetails**

PRIMARY KEY: examDtlID

FOREIGN KEY: examCatID, courseBatchID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| examDtlID\* | Varchar(50) | Exam Details ID |
| examCatID# | Varchar(50) | Exam Category ID |
| conductedOn | DateTime | Conducted On |
| NoOfAtn | Numeric(20) | No. Students Attended |
| courseBatchID# | Varchar(50) | Course Batch ID |

**13. examPaper**

PRIMARY KEY: paperID

FOREIGN KEY: examDtlID, questionID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| paperID\* | Varchar(50) | Question Paper ID |
| examDtlID# | Varchar(50) | Exam Details ID |
| questionID# | Varchar(50) | Question ID |

**14.studentsResult**

PRIMARY KEY: stdResultID

FOREIGN KEY: stdID, examDtlID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| stdResultID\* | Varchar(50) | Student Result ID |
| examDtlID# | Varchar(50) | Exam Details ID |
| markObtained | Numeric(20,22) | Mark Scored |
| stdID# | Varchar(50) | Student’s ID |

**15. courseDailyReport**

PRIMARY KEY: crsDailyRpt

FOREIGN KEY: crsSyID, crsLivSchID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| crsDailyRpt\* | Varchar(50) | Course Daily Report ID |
| Date | DateTime | Date |
| activeStd | Varchar(50) | No. Of Active Students |
| crsSyID# | Varchar(50) | Portion Covered |
| crsLivSchID# | Varchar(50) | crsLivSchID |

**16.discussionBoard**

PRIMARY KEY: dbID

FOREIGN KEY: coursed, stdID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| dbID\* | Varchar(50) | Discussion Board ID |
| courseId# | Varchar(50) | Course ID |
| Query | Varchar(500) | Query asked |
| dateTime | DateTime | Date And Time |
| stdID# | Varchar(50) | Student Asked Query |

**17. dbResponse**

PRIMARY KEY: DBRID

FOREIGN KEY: dbID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| DBRID\* | Varchar(50) | Discussion Board Response |
| dbID# | Varchar(50) | Discussion Board ID |
| userEmail | Varchar(50) | User E-Mail |
| Response | Varchar(500) | Response To Query |

**18.message**

PRIMARY KEY: msgNo

FOREIGN KEY:

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| msgNo\* | Varchar(50) | Message No |
| msgType | Varchar(50) | Message Type |
| dateTime | Varchar(50) | Date and Time |
| Sender | Varchar(50) | Sender |
| Receiver | Varchar(50) | Receiver |
| msgTxt | Varchar(50) | Message Text |
| Status | Varchar(50) | Status |

**19.MsgImage**

PRIMARY KEY: msgImgID

FOREIGN KEY: msgNo

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| msgImgID\* | Varchar(50) | Message Image ID |
| msgNo# | Varchar(50) | Message No. |
| msgDescription | Varchar(500) | Image Description |
| Path | Varchar(50) | Path Of Image |

**20.notification**

PRIMARY KEY: notificationID

FOREIGN KEY:

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| notificationID | Varchar(50) | Notification ID |
| notifiDateTime | DateTime | Notify Date and Time |
| Type | Varchar(50) | Type Of Notification |
| Sender | Varchar(50) | Notify by |
| Receiver | Varchar(50) | Notify To |
| Text | Varchar(50) | Text |
| Status | Varchar(50) | Notification Status |

**21.notificationImgVid**

PRIMARY KEY: notImgID

FOREIGN KEY: notificationID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| notImgID\* | Varchar(50) | Notification Image ID |
| notificationID# | Varchar(50) | Notification ID |
| Path | Varchar(50) | Path |

**22.courseStudents**

PRIMARY KEY: crsStdID

FOREIGN KEY: courseBatchID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| crsStdID\* | Varchar(50) | Course Student Have |
| courseBatchID# | Varchar(50) | Course Batch ID |
| stdID | Varchar(50) | Students ID |

**23.dailyStdReport**

PRIMARY KEY: dailyStdRpt

FOREIGN KEY: stdID, crsDailyRptID

|  |  |  |
| --- | --- | --- |
| **FIELDS** | **DATATYPE** | **DESCRIPTION** |
| dailyStdRpt\* | Varchar(50) | Daily Report ID |
| stdID# | Varchar(50) | Student’s ID |
| crsDailyRptID# | Varchar(50) | Course daily Report ID |
| activeAt | DateTime | Active Time |
| Duration | Numeric(20,22) | Time Active For |
| pointScored | Numeric(10) | Point Scored |