**1 INTRODUCTION**

The title of the project is “Multi School Integration”. This project will handle whole the activities of the school. SMS has most of the facilities that modern school requires to computerize its day- to-day jobs. It provides facilities to keep the records of student, marks, teaching and non-teaching staff with all their required details along with all required transactions handling. It has facilities to generate various types of reports, which are required by the management during normal business operation to operate the business effectively.

|  |  |
| --- | --- |
|  | **2 SYSTEM ANALYSIS** |

**2.1 FEASIBILITY STUDY:**

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
* Economical Feasibility
* Operation Feasibility

**2.1.1 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.

**2.1.2 Economic 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.

**2.1.3 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 SYSTEM SPECIFICATIONS**

**3.1 HARDWARE SPECIFICATION**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, a hardware requirements list is often accompanied by a hardware compatibility list (HCL). An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application.

Processor : Intel Duel core

RAM : 2 GB

Hard Disk : 10 GB

3.2 SOFTWARE SPECIFICATION

It is a complete description of the behavior of a system to be developed and It includes a set of use cases that describe all the interactions the user will have with the software. In addition to use cases, the SRS also contains non-functional requirements.

Operating System : Windows 7

Front End : Python with Django

Back End : SQLite

**4 SELECTED SOFTWARE DESCRIPTION**

**4.1. HYPER TEXT MARKUP LANGUAGE**

HTML stands for "Hypertext Markup Language". HTML is a SGML (Standard Generalized Markup Language) application widely used to create web pages. It is basically a formatting language and not a programming language.HTML is a language that is easy to write, easy to understand and highly portable. HTML is not a compiled language and is directly interpreted by a browser.HTML is the set of instructions. Each instruction is called as an element or Markup. It is used to structure and format documents for presentation on the web. HTML enhances ASCII files with markup tags that permit the display of a variety of fonts, images, and highlighting options. It also designates structural elements such as headers, lists, and paragraphs, and provides hypertext links to other documents on the Internet.

**4.1.1 Interactive HTML:**

Html TagThe first and last tags in a document should always be the HTML tags. These are the tags that tell a Web browser where the HTML in your document begins and ends. The absolute most basic of all possible Web documents is:

<Html >

</Html>

That's it. If we were to load such a page into a Web browser, it wouldn't do anything except give us a blank screen, but it is technically a valid Web page.

**Head Tag** The HEAD tags contain all of the document's header information. When I say "header," I don't mean what appears at the top of the browser window, but things like the document title and so on. Speaking of which...

**Body Tag** BODY comes after the HEAD structure. Between the BODY tags, all of the stuff that gets displayed in the browser window is found. All of the text, the graphics, and links, and so on these things occur between the BODY tags.

**Forms Tag** Forms provide a unique feature to HTML. Forms allow you to collect data from the end user and return that data to an executable code. The <FORM> element is used to start a form. The <FORM> element specifies what program to run when the form is submitted and how the data is to be transferred.

**Attributes Tag** ACTION-The ACTION attribute specifies what program or HTML file is to be called when the submit button is pressed. The ACTION is specified as a URL.

**Method**The METHOD attribute specifies the protocol to be used when the client sends data to the server. There are two methods. GET (the default) or POST. Using GET method data is attached to the URL mentioned in ACTION attribute. The data that can be sent here is limited here. Where as in POST the data is posted through environment variables and unlimited amounts of data can be sent.

**Name** the NAME attribute specifies the name of the form with which the elements of the form can be referred to later.

***The <INPUT> Element*** The<INPUT> tag provides some type of data entry in the form depending on the value of its type attribute.

**Attributes** HREF - HREF stands for "Hypertext Reference," which is another way of saying, "The location of the file I want to load." Most anchors are in the form <A HREF="URL">, where URL is the location of the resource to which you want the link to point. So the HREF attribute of the Anchor element specifies a URL.

**4.2. JAVA SCRIPT**

Java Script is Netscape’s cross–platform, object-based scripting language for client server application. JavaScript is mainly used as a client side scripting language. This means that JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it's up to the browser to do something with it. JavaScript can be used in other contexts than a Web browser. Netscape created server-side JavaScript as a CGI-language that can do roughly the same as Perl or ASP.

Fortunately most browsers can handle JavaScript nowadays, but of course some browsers do not support some bits of script.

**Types of Java Script**

a. Navigator Java Script also called client-side Java Script.

b. Live Wire Java Script also called server-side Java Script.

**Features of JavaScript (JS)**

1. Browser interprets JavaScript.
2. JavaScript is object based and uses built-in, extensible objects and have no classesor inheritance
3. JavaSript is loosely typed language
4. In JavaScript object reference are checked at runtime
5. JavaScript is designed to supplement the capabilities of HTML with script that are capable of responding to web pages events. JSP has access to some extent of aspects of the web browser window.
6. JavaScript control browser and content but cannot draw graphics or perform networking.

**The Client-Side JavaScript also has the following features**

* Controls Document’s appearance and content
* Control the browser
* Interact with the HTML forms
* Interact with the user
* Read and write client state with cookies
* Server- Side JavaScript Features
* Embedded in HTML page
* Executed at the server
* Pre-complied for faster response
* Access to Server-side objects
* Encapsulation of the request
  1. **JQuery:**

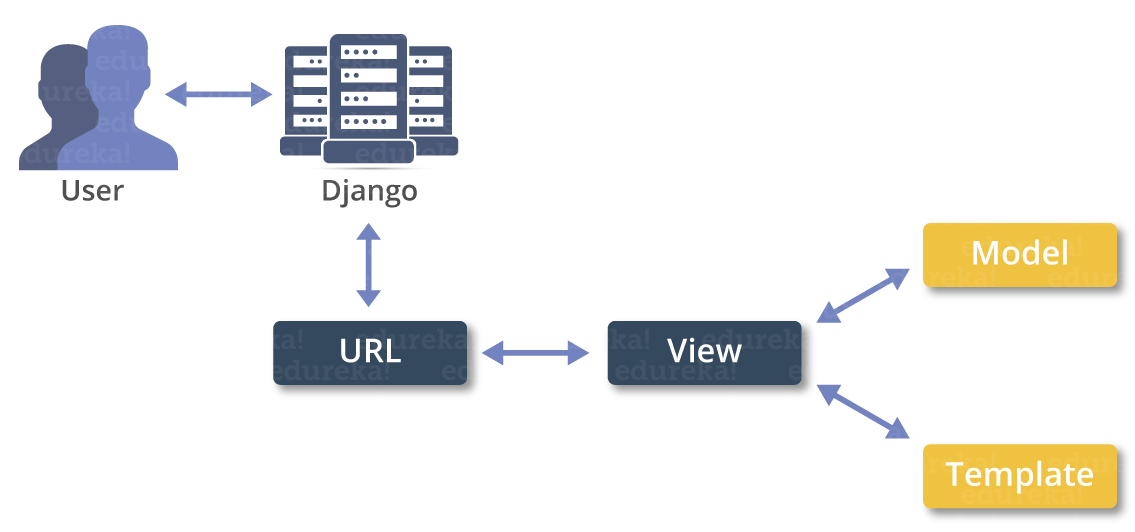
JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript. Support from our corporate members makes it possible for the jQuery Foundation to continue our work on our JavaScript libraries and pushing the open web forward with events and participation in the standards process.

**4.4 Python**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* + 1. **Architecture of Django:**



**4.5 Database:**

A database management (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data,and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle,DB2,Microsoft SQL Server, Microsoft Access, Firebird, MySQL, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems.

Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

**4.5.1 Description:**

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:

* A modeling language to define the schema of each database hosted in the DBMS, according to the DBMS data model.
  + The four most common types of organizations are the hierarchical, network, relational and object models. Inverted lists and other methods are also used.
  + The dominant model in use today is the adhoc one embedded in sql, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
* Data structures optimized to deal with very large amounts of data stored on a permanent data storage device
* A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.
* It also controls the security of the database.
* Data security prevents unauthorized users from viewing or updating the database. using passwords, users are allowed access

to the entire database or subsets of it called sub-schemas. For example, an employee database can contain all the data about an individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and medical data.

If the dbms provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

* A transcationmechanism that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses and faults.
* It also maintains the integrity of the data in the database.
* The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time, The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers can be entered into the database. See ACID properties for more information.

The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data.

**4.6 SQL:**

SQL is a database computer language designed for the retrieval and management of data in relational database. SQL stands for Structured Query Language. In the relational model, data is stored in structures called relations or tables. SQL, statements are issued for the purpose of:

**4.6.1 Data Definition**:

Defining tables and structures in the database

**4.6.2 Data Manipulation:**

Used to manipulate the data within those schema objects

A schema is a collection of database objects that can include: tables, views, indexes and sequences

List of SQL statements that can be issued against an Oracle database schema are:

* **ALTER –** Change an existing table, view or index definition
* **AUDIT –** Track the changes made to a table
* **COMMENT –**Create new database objects such as tables or views
* **DELETE –** Delete rows from a database tables
* **DROP –** Drop a database object such as table, view or index
* **GRANT –** Allow another user to access database objects such as tables or views
* **INSERT –** Insert new data into a database table
* **NO AUDIT –** Turn off the auditing function
* **REVOKE –** Disallow a user access to database objects such as tables and views
* **ROLLBACK –** Undo any recent changes to the database
* **SELECT –** Retrieve data from a database table
* **TRUNCATE-** Delete all rows from a database table

**5 PROJECT DESCRIPTION**

**5.1 OVERVIEW OF THE PROJECT**

A school management system includes several tasks such as student’s registration, record keeping of student mark details, generating report cards, and others. A school management software is a complete solution that caters the needs of school administration process.

In the current system, we need to keep a number of records related to the student and retrieving the records as per need is a tedious job. The school management might require customized reports of students and search for the entire data to generate one is highly impossible and a school management software must be capable of generating customized reports as in when needed. In the proposed system, we offer the provision for generating reports as per the requirements. There is a separate module for reports displaying various options underneath it like mark reports, search for student, staff reports etc. You can click on any of the options to create your customized report.

**PROBLEM STATEMENT**

In the current system, we need to keep a number of records related to the student and retrieving the records as per need is a tedious job. The school management might require customized reports of students and search for the entire data to generate one is highly impossible and a school management software must be capable of generating customized reports as in when needed.

**PURPOSE OF THE PROJECT:**

Using my project any one can get information from any location in the world. Another major purpose of the project is providing proper information about student mark details.

**SCOPE OF THE PROJECT:**

we offer the provision for generating reports as per the requirements. There is a separate module for reports displaying various options underneath it like mark reports, search for student, staff reports etc. You can click on any of the options to create your customized report.

**5.2 MODULE SPECIFICATION**

The modules of the project are: –

1. Login Module
2. Registration Module
3. Staff Module
4. Mark Entry Module

**5.2.1 Login Module: –**

This will help users to login into the system using institute id and password. A user who has the valid id and password can only login to their respective accounts. It will help the authentication of the user who enters the system.

**5.2.2 Registration Module: –**

This module the student will get registered as it is new in the educational institute. It will be formed like structure where all the student details will be filled. It will have the fields regarding their personal information like date of birth and address.

**5.2.3 Mark Entry Module: –**

The main function of this module is record the marks of a particular student in a institute. The staff can add edit and delete marks for the student.

**5.2.4 Staff Management:**

Details of staff and non-teaching staff is maintained. One can maintain details such as name, phone number and subject details. The main function of this module is add ,edit and delete Staff details

**5.3 SYSTEM DESIGN**

**5.3.1 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.

**5.3.2 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.

**FIRST NORMAL FORM**:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

**SECOND NORMAL FORM**:

A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1. Primary key is a not a composite primary key
2. No non key attributes are present
3. Every non key attribute is fully functionally dependent on full set of primary key.

**THIRD NORMAL FORM**:

A relation is said to be in third normal form if their exits no transitive dependencies.

**Transitive Dependency**: If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

**E-R Diagram:**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. An **entity-relationship model** (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database,



**5.3.3 DATA FLOW DIAGRAMS**

A data flow diagram is 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.

**DFD SYMBOLS:**

In the DFD, there are four symbols

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

Process that transforms data flow.

Source or Destination of data

Data flow

Data Store

**CONSTRUCTING A DFD:**

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

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

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

**SAILENT FEATURES OF DFD’S**

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

***TYPES OF DATA FLOW DIAGRAMS***

1. Current Physical
2. Current Logical
3. New Logical
4. New Physical

**CURRENT PHYSICAL:**

In Current Physical DFD process label include the name of people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly data flows and data stores are often labels with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

**CURRENT LOGICAL:**

The physical aspects at the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transforms them regardless of actual physical form.

**NEW LOGICAL**:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from current logical model while having additional functions, absolute function removal and inefficient flows recognized.

**NEW PHYSICAL:**

The new physical represents only the physical implementation of the new system.

**RULES GOVERNING THE DFD’S**

**PROCESS**

1. No process can have only outputs.
2. No process can have only inputs. If an object has only inputs than it must be a sink.
3. A process has a verb phrase label.

**DATA STORE**

1. Data cannot move directly from one data store to another data store, a process must move data.
2. Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
3. A data store has a noun phrase label.

**SOURCE OR SINK**

The origin and /or destination of data.

1. Data cannot move direly from a source to sink it must be moved by a process
2. A source and /or sink has a noun phrase land

**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.





**Second Level DFD**

**Use Case Diagrams :**



**5.4 INPUT DESIGN**

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

* To produce 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 States:**

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 input. Inputs can be categorized as follows:

* External Inputs which are prime inputs for the system.
* Internal Inputs, which are user communications with the systems.
* 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.

**5.5 OUTPUT DESIGN:**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide 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 with User Interface.

**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.

For Example

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

**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.

1. **SYSTEM TESTING AND IMPLEMENTATION**

**6.1 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.

**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.

UNIT TESTING

MODULE TESTING

SUB-SYSTEM TESING

SYSTEM TESTING

ACCEPTANCE TESTING

Component Testing

Integration Testing

User Testing

**6.1.1 UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

**1. WHITE BOX TESTING**

This type of testing ensures that

* All independent paths have been exercised at least once
* All logical decisions have been exercised on their true and false sides
* All loops are executed at their boundaries and within their operational bounds
* All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

**2. BASIC PATH TESTING**

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

V(G)=E-N+2 or

V(G)=P+1 or

V(G)=Number Of Regions

Where V(G) is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

**3. CONDITIONAL TESTING**

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

**4. DATA FLOW TESTING**

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

**5. LOOP TESTING**

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

All the loops were tested at their limits, just above them and just below them.

All the loops were skipped at least once.

For nested loops test the inner most loop first and then work outwards.

For concatenated loops the values of dependent loops were set with the help of connected loop.

Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

**6.1.2 INTEGRATION TESTING**

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps

## 6.1.3 VALIDATION TESTING

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements.Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

**6.1.4 BLACK BOX TESTING**

Black-box testing is a method of [software testing](https://en.wikipedia.org/wiki/Software_testing) that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing [unit](https://en.wikipedia.org/wiki/Unit_test), [integration](https://en.wikipedia.org/wiki/Integration_testing), [system](https://en.wikipedia.org/wiki/System_testing) and [acceptance](https://en.wikipedia.org/wiki/Acceptance_test). It is sometimes referred to as specification-based testing.

**6.2 TEST CASES**

**SYSTEM SECURITY**

# 6.2.1 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.

**6.2.2 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.
* Using server side validation, constraints on several restricted operations are imposed.

**CONCLUSION**

**School Management System Project**is to avoid  manual problems and also documentation storage problem we can’t maintain long period data that’s why we used computerized system to overcome all problem related to school’s data storing and other arias. All modules in the system have been tested with valid data and invalid data and everything work successfully. This work is started with an ambition to computerize the organization. After the detailed analysis, this project was developed and implemented. This project is successfully completed and it is found to be working well.

**FUTURE ENHANCEMENT**

Changes in the software engineering technology are indeed rapid. This project has many merits and demerits. The demerits can be overcome by future enhancements to the system. The project developed deals only with student, staff, mark details etc., but it also has another major sections .That sections that are attendance , fees etc., are the problems that is not given a solution in this. The developed project can be altered to fulfill the enhancements. These may be included in this system in the future to make this project as a 100% perfect and efficient system.