

CONTENTS

1. INTRODUCTION	05
2. SYSTEM ANALYSIS	06
2.1 PROJECT IDENTIFICATION OF NEED	06
2.2 PRELIMINARY INVESTIGATION	06
2.2.1 STUDY OF EXISTING SYSTEM	06
2.2.2 PROPOSED SYSTEM	07
2.2.3 ADVANTAGES OF PROPOSED SYSTEM	07
3. FEASIBILITY STUDY	09
3.1 ECONOMIC FEASIBILITY	09
3.2 TECHNICAL FEASIBILITY	09
3.3 OPERATIONAL FEASIBILITY	10
4. SOFTWARE ENGINEERING PARADIGMS APPLIED	11
5. SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATION	12
5.1 REQUIREMENT SPECIFICATION	12
5.1.1 ANALYSIS OF FACTUAL DATA	12
5.1.2 IDENTIFICATION OF ESSENTIAL REQUIREMENT	12
5.1.3 SELECTION OF REQUIREMENT STRATEGIES	13
5.2 DEFINITION OF INPUT REQUIREMENTS	13
5.3 DEFINITION OF PROCESSING REQUIREMENTS	14
5.4 DEFINITION OF OUTPUT REQUIREMENTS	14
5.5 OBJECTIVE OF SRS	15

5.6 OVERVIEW OF SRS	15
5.6.1 OVERALL DESCRIPTION	15
5.6.2 SPECIFIC REQUIREMENTS	15
5.7 HARDWARE CONFIGURATION	16
5.8 SOFTWARE CONFIGURATION	16
6. SYSTEM DESIGN	17
6.1 DESIGN METHODOLOGY	17
6.1.1 LOGICAL DESIGN	17
6.1.2 PHYSICAL DESIGN	18
6.1.3 MODULAR DESIGN	18
6.2 INPUT DESIGN	18
6.3 OUTPUT DESIGN	19
6.4 DATABASE DESIGN	19
6.4.1 NORMALIZATION	20
6.4.2 TABLES	21
1. PERSONAL DETAILS	21
2. LOGIN	22
3. EXAM	22
4. COURSE	23
5. ADDRESS	23
6.4.3 DATAFLOW DIAGRAMS	24
Level 0	24
Level 1	24
Level 2	25
Level 3	26
7. CODING	27
7.1 CODE EFFICIENCY	27
7.1.1 CORRECTNESS	27

7.1.2 RELIABILITY	28
7.1.3 ROBUSTNESS	28
7.1.4 DEVELOPMENT EFFICIENCY	28
7.1.5 EXECUTION EFFICIENCY	28
7.2 TECHNOLOGY SPECIFICATIONS	28
7.2.1 PHP (Hyper text Preprocessor)	28
7.2.2 Macromedia Dream Weaver	30
7.2.3 My SQL	30
7.3 OPTIMIZATION OF CODE	30
7.4 VALIDATION CHECKS	30
8. IMPLEMENTATION AND MAINTANENCE	31
8.1 IMPLEMENTATION PLANS	31
8.2 CHANGEOVER METHOD	31
8.3 TESTING	31
8.3.1 TESTING OBJECTIVES	32
8.3.2 UNIT TESTING	32
8.3.3 SYSTEM TESTING	33
8.3.4 INTEGRATION TESTING	33
8.3.5 WHITE BOX TESTING	33
8.3.6 BLACK BOX TESTING	34
8.4 SYSTEM MAINTENANCE	34
9. SYSTEM SECURITY MEASURES	35
9.1 TECHNICAL SECURITY MEASURES	35
9.2 INFRASTRUCTURAL SECURITY MEASURES	35
9.3 ORGNISATIONAL SECURITY MEASURES	35
9.4 PERSONAL-RELATED SECURITY MEASURES	35
10. COST ESTIMATION OF PROJECT	36
10.1 PERT CHART	36

10.2 GANTT CHART	37
11. FUTURE SCOPE	38
12. CONCLUSION	38
13. BIBLIOGRAPHY	39
14. APPENDICES	40

1.INTRODUCTION

Online Admission System is aimed at developing an online admission application for a college. This system is an online system that can be accessed throughout the organization and outside as well with proper login provided. Our system has two type of accessing modes, administrator and user. Student management system is managed by an administrator. It is the job of the administrator to admit and monitor the whole process. When a user log in to the system. He would only view details of the student. He can't perform any changes .The system has two modules. They are

User

Administrator

Students logging is to apply for the course by filling an applycation form provided by online. College principal/administrator logging in may also access/search information put up by the students.

2. SYSTEM ANALYSIS

System analysis aims at establishing requests for the system to be acquired, developed and installed. It involves studying and analyzing the ways of an organization currently processing the data to produce information. Analyzing the problem thoroughly forms the vital part of the system study. In system analysis, prevailing situation of problem carefully examined by breaking them into sub problems. Problematic areas are identified and information is collected. Data gathering is essential to any analysis of requests. It is necessary that this analysis familiarizes the designer with objectives, activities and the function of the organization in which the system is to be implemented.

2.1 PROJECT IDENTIFICATION OF NEED

As the strength of the students is increasing at a tremendous speed, manual maintenance of student admission is very difficult. Hence, the need for online admission is inevitable. In case of manual system they need a lot of time, manpower etc. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes.

2.2 PRELIMINARY INVESTIGATION

2.2.1 STUDY OF EXISTING SYSTEM

Today in colleges student details are entered manually. The student details in separate records are tedious task. Referring to all these records and updating is needed. There is a chance for more manual errors.

1. When the student comes in college.
2. First of all, he/she takes admission form from reception.
3. Fills it and submits it into office.
4. Filled form is first checked with documents like merit list and details came from university and verified by an official person, if there is any mistake then it is corrected.
5. At the time of submission of it the fees is deposited by the candidate.
6. At the time of submission of admission form admission no. is assigned to the candidate by the institute.

7. Candidate gets the receipt of fees deposition.

Disadvantages of Present System:-

1. Require much man power i.e. much efforts, much cost and hard to operate and maintain.
2. Since, all the work is done in papers so it is very hard to locate a particular student record when it is required.

2.2.2 PROPOSED SYSTEM

The main goal of the system is to automate the process carried out in the organization with improved performance and realize the vision of paperless admission. Some of the goals of the system are listed below:

- ☐ Manage large number of student details.
- ☐ Manage all details of student who registered for the course
- ☐ Create student accounts and maintain the data's effectively.
- ☐ View all the details of the students.
- ☐ Reduce the work load in interview the students for selection
- ☐ Activities like updating, modification, deletion of records should be easier.

2.2.3 ADVANTAGES OF PROPOSED SYSTEM

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the objectives of the proposed system:

- ☐ *Reach to geographically scattered students.* One of the important objectives of the admission system is communicate with all the students scattered geographically.
- ☐ *Reducing time in activities.* Reduce the time taken process the applications of students, admitting a student, conducting the online examination, verify student marks, and send call letters to selected students.
- ☐ *Centralized data handling.* Transfer the data smoothly to all the departments involved and handle the data centralized way.

- ☐ *Paperless admission with reduced manpower.* Reduce the manpower needed to perform all the admission and administration task by reducing the paper works needed.
- ☐ *Cost cutting.* Reduce the cost involved in the admission process.
- ☐ *Operational efficiency.* Improve the operational efficiency by improving the quality of the process.

3. FEASIBILITY STUDY

3.1 ECONOMIC FEASIBILITY

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions are made to design and implement the system.

This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases.

It is economically feasible, it will only require a single operator to operate the system, who is responsible for entering the data into the database via a user interface provided to him, who can also be able to show all the data in html tabular form so to provide information regarding the students who are either taken admission or to take admission, since it requires only a single person to operate the whole system thus reduces the cost to operate the system.

In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving a lot of time.

3.2 TECHNICAL FEASIBILITY

Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. It is technically feasible, since the whole system is designed into the latest technologies like PHP and SQL Server which are the most recent technologies to develop web based systems and design databases.

The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very

high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

3.3 OPERATIONAL FEASIBILITY

It is Operational feasible, since the system is providing a attractive user interface to the operator/end user, so he feel very easy to work onto it. Response to operator/end user is very fast and very good. Since, as we mentioned above that it requires much less amount of cost, it uses computer work so it is very fast to operate and it is very easy for user to work on it.

4. SOFTWARE ENGINEERING PARADIGMS APPLIED

The two main programming approaches are top-down approach and bottom-up approaches. Here while doing the software for online placement system; we use the bottom up approach.

In bottom up method, the sub modules are designed and developed and these are assembled together to get the main system. The individual modules are coded and debugged with example data given checks ready for the major project design. The individual module transaction processing can be checked thoroughly so that minute mistakes and correction can be passed from one module to other when required and these requisition are also considered at the development stage of code module. These checked and corrected modules will be integrated to make the entire system. This project also followed the bottom up design. The sub modules are prepared very keenly and minute notations are made.

In the online admission system, the sub modules are designed first, these sub modules are integrated into main modules. Other modules also designed in the same fashion. Each module is tested at the beginning when they are integrated with the sub-modules

These checked and corrected modules are integrated to make the main system, i.e. Online Admission System.

5. SOFTWARE AND HARDWARE REQUIREMENT SPECIFICATION

5.1 REQUIREMENT SPECIFICATION

The final output is the requirements specification document (SRS). For smaller problems or problems that can easily be comprehended; the specification activity might come after the entire analysis is complete. However, it is more likely that problem analysis and specification are done concurrently. All the information for specification activity as following the analysis activity. The transition from analysis to specification should also not be expected to be straightforward, even if some formal modeling is used during analysis. Essentially, what passes from requirements analysis activity to the specification activity is the knowledge acquired about the system. The modeling is essentially a tool to help obtain a thorough and complete knowledge about the proposed system.

5.1.1 ANALYSIS OF FACTUAL DATA

Analysis of data is a process of inspecting, cleaning, transforming, and modeling [data](#) with the goal of highlighting useful [information](#), suggesting conclusions, and supporting decision making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, in different business, science, and social science domains.

[Data mining](#) is a particular data analysis technique that focuses on modeling and knowledge discovery for predictive rather than purely descriptive purposes.

5.1.2 IDENTIFICATION OF ESSENTIAL REQUIREMENT

Identification of essential requirement is an important task in developing the project. In this system the essential requirements are identified through surveying. By surveying, the important needs of the user in our website are known. In the surveying, the different possibilities of tour information that have to be included in the website is given by questionnaire.

Questions included like:

- Need to change the current admission formalities.
- Is it advantageous?

5.1.3 SELECTION OF REQUIREMENT STRATEGIES

From the survey analysis graph it is clear that which are all the requirements that the user requires the most. It is decided to include the required information and omit the less priority ones.

5.2 DEFINITION OF INPUT REQUIREMENTS

5.2.1 REGISTRATION & LOGIN SYSTEM

Applicants will carry out their own registration, providing the system with a way to associate a user to their application(s). This will enable the system to display personalised information when the user logs in and certain information, such as name and address, to be added to each application automatically. Giving each student a specific ID will also allow a user to apply to a number of courses, while giving the system a way to prevent unnecessary duplication of applications. Requiring a registration process will also add greater security to the system, as once a user has logged in with their username and password, they will be the only person able to update their information and the only person to make applications on their behalf.

5.2.2 APPLICATION SYSTEM

The application process will be as straightforward as possible, using an intuitive form layout, with the necessary information being completed in stages. The system will monitor this and not allow a student to apply to a single course more than once, but will allow them to apply to multiple courses in the same college

5.2.3 VIEW STATUS

When an application has been completed by a student, they will be able to log in to the system at any time and view its status

5.2.4 UPDATE DETAILS

Applicants, will also be able to update their application details. Forms, much like the ones used to register and initially apply, will be the means of inputting the new details and will contain the previous information as a starting point. The only time an application will be locked for editing will be when it has been submitted for admission process that is after date of last submission, after which point the application will no longer be accessible by the user.

5.3 DEFINITION OF PROCESSING REQUIREMENTS

The user interface for this system will have to be simple and clear. Most importantly, the pages must be easy to read, easy to understand and accessible. The colour scheme should be appropriate to provide familiarity with the university and there should be no contrast issues.

There are many functions the system can perform and these must be logically grouped or displayed in an intuitive order to allow the user to perform tasks quickly and efficiently, without getting lost in excessive amounts of text. The system must also display a large amount of information and to avoid confusion this must be displayed in categories or in different pages. Furthermore, a small amount of information may be displayed initially, for example with a certain limit on date or amount, and the ability to view more in depth information on the subject should be apparent.

The different information displays and functionality objects should be individually distinguishable, allowing the user to navigate through recognition, rather than recall. In addition, each function must provide the ability to cancel, leaving the user with the ability to rectify mistakes, and every page should include the ability to return to a central location of the system, ensuring that the user does not get lost within the system with no convenient way to navigate.

The system will provide different views for different users, allowing multiple access levels. For example, a student will only be able to see their own details and details of their applications, whereas an administrator will be able to view all users, applications and statistics and will have many more privileges. Being an online system, it will naturally be viewable from any computer with an internet connection, allowing admissions from home, for example. This will provide far more accessibility than if it were written in a language with only limited online capability as any computer is a potential work station, rather than relying on the program being installed.

5.4 DEFINITION OF OUTPUT REQUIREMENTS

The most important function is to make the short list of student who got admission under the circumstances made by the college

In some cases, decisions about an application will be simple, given that the application might be exceptionally good or exceptionally bad. If, however, an application is

similar to other, previous applications, the tutor may have a more difficult decision to make and inconsistencies may be introduced. Using the automatic ranking of applications a tutor will be able to see a list of applications with a similar ranking. This list will have a default length of 5, for example, but this will be extendible if more comparisons are needed, and the list will include applications of the same rank as well as slightly higher and lower ranks.

5.5 OBJECTIVE OF SRS

The objective of this SRS document is to specify software requirements of the Online Admission for the college. It is intended to be a complete specification of what functionality the admission provides. The main purpose of the system is to automate the task carried out by different peoples in the organization to perform the student admission. Specific design and implementation details will be specified in a future document.

5.6 OVERVIEW OF SRS

SRS will include two sections.

Overall Description will describe major components of the system, interconnection and external interfaces.

Specific Requirements will describe the functions of actors, their role in the system and constraints.

5.6.1 OVERALL DESCRIPTION

The SRS document will give further details on the overall product description, including the hardware, software, and communications interfaces, product functions, user characteristics, and any assumptions that will be made.

5.6.2 SPECIFIC REQUIREMENTS

The SRS document will also include the specific requirements needed. These will include the functions, performance, design, and software attributes. This document is organized in a logical manner and is easy to follow. Readers should refer to the table of contents, appendices, or index if looking for something in specific. Otherwise, reading this document from start to finish will start with a vague description and get more specific and detailed as changing sections and reading further.

5.7 HARDWARE CONFIGURATION

- System : Multimedia PC
- Processor : Pentium 4 or above
- Memory : 512MB RAM
- Hard Disk : 80GB or above
- Keyboard : 104 standards
- Monitor : SVGA
- Modem : Dial up/Broadband

5.8 SOFTWARE CONFIGURATION

- Front end : PHP, HTML, CSS, JavaScript
- Back end : MYSQL
- Operating System : Windows ,LINUX

6. SYSTEM DESIGN

The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The first step in system designing is to determine how the output is to be produced and in what format. Samples of the output and input are also presented. In the second step, input data and master files are to be designed to meet requirement of the proposed output. The processing phases are handled through program construction and testing, including a list of the programs needed to meet the system's objectives and complete documentation.

6. 1 DESIGN METHODOLOGY

System design is the solution to the creation of a new system. This phase is composed of several systems. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design.

During logical design phase the analyst describes inputs (sources), out puts (destinations), databases (data sores) and procedures (data flows) all in a format that meats the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design.

The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

6.1.1 LOGICAL DESIGN

Logical design of an information system shows the major features and also how they are related to one another. The first step of the system design is to design logical design elements. This is the most creative and challenging phase and important too. Design of proposed system produces the details of the state how the system will meet the requirements

identified during the system analysis that is, in the design phase we have to find how to solve the difficulties faced by the existing system. The logical design of the proposed system should include the details that contain how the solutions can be implemented. It also specifies how the database is to be built for storing and retrieving data, what kind of reports are to be created and what are the inputs to be given to the system. The logical design includes input design, output design, and database design and physical design

6.1.2 PHYSICAL DESIGN

The process of developing the program software is referred to as physical design. We have to design the process by identifying reports and the other outputs the system will produce. Coding the program for each module with its logic is performed in this step. Proper software specification is also done in this step.

6.1.3 MODULAR DESIGN

A software system is always divided into several sub systems that makes it easier for the development. A software system that is structured into several subsystems makes it easy for the development and testing. The different subsystems are known as the modules and the process of dividing an entire system into subsystems is known as modularization or decomposition.

A system cannot be decomposed into several subsystems in any way. There must some logical barrier, which facilitates the separation of each module. The separation must be simple but yet must be effective so that the development is not affected.

The system under consideration has been divided into several modules taking in consideration the above-mentioned criteria. The different modules are

- 1.user module
- 2..administrator module

6.2 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to

read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple.

The system needs the data regarding the asset items, depreciation rates, asset transfer, physical verification for various validation, checking, calculation and report generation.. The error raising method is also included in the software, which helps to raise error message while wrong entry of input is done. So in input design the following things are considered.

- What data should be given as input?
- How the data should be arranged or coded?
- Methods for preparing input validations and steps to follow when error occur
- The samples of screen layout are given in the appendix.

6.3 OUTPUT DESIGN

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So while designing output the following things are to be considered.

- Determine what information to present
- Arrange the presentation of information in an acceptable format
- Decide how to distribute the output to intended receipts

Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy. The options for the output reports are given in the appendix.

6.4 DATABASE DESIGN

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. DBMS allow data to be protected and organized separately from other resources. Database is an integrated collection of data.

The most significant form of data as seen by the programmers is data as stored on the direct access storage devices. This is the difference between logical and physical data.

Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information.

The organization of data in database aims to achieve three major objectives: -

- Data integration.
- Data integrity.
- Data independence.

The proposed system stores the information relevant for processing in the MS SQL SERVER database. This database contains tables, where each table corresponds to one particular type of information. Each piece of information in table is called a field or column. A table also contains records, which is a set of fields. All records in a table have the same set of fields with different information. There are primary key fields that uniquely identify a record in a table. There are also fields that contain primary key from another table called foreign keys.

6.4.1 NORMALIZATION

Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion and updating anomalies. All the tables have been normalized up to the third normal form. In short the rules for each of the three normal forms are as below.

- First normal form

A relation is said to be in 1NF if all the underlying domain of attributes contain simple individual values.

- Second normal form

The 2NF is based on the concept of full functional dependency. A relation is said to be in 2NF if and only if it is in 1NF and every non-key attribute is fully functionally dependent on candidate key of the table.

- Third normal form

The 3NF is based on the concept of transitive dependency. A relation in 2NF is said to be in 3NF if every non-key attribute is non-transitively

6.4.2 TABLES

1. PERSONAL DETAILS

FIELDNAME	DATATYPE
<u>regid</u>	Varchar()
name	Varchar()
father	Varchar()
age	Integer
gender	Varchar()
religion	Varchar()
caste	Varchar()
day	Integer
Month	Integer
year	Integer

2. LOGIN

FIELDNAME	DATATYPE
<u>Regid</u>	Varchar()
Password	Varchar()
Type	Varchar()
Flag	Integer

3. EXAM

FIELDNAME	DATATYPE
<u>Regid</u>	Varchar()
Eqxam	Varchar()
Rollno	Integer
Passyear	Integer
Mark	Integer
Total	Integer
Percent	Integer

4. COURSE

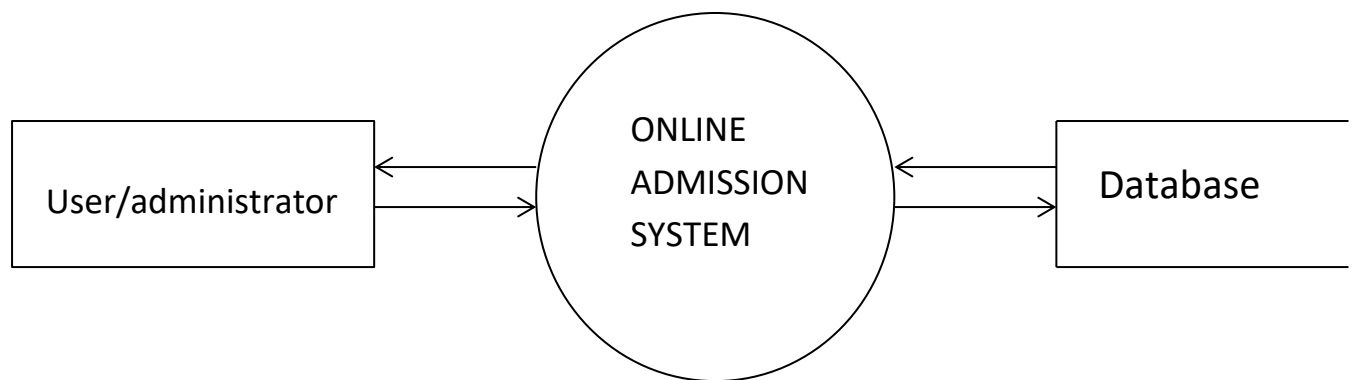
FIELDNAME	DATATYPE
<u>regid</u>	Varchar()
course	Varchar()

5. ADDRESS

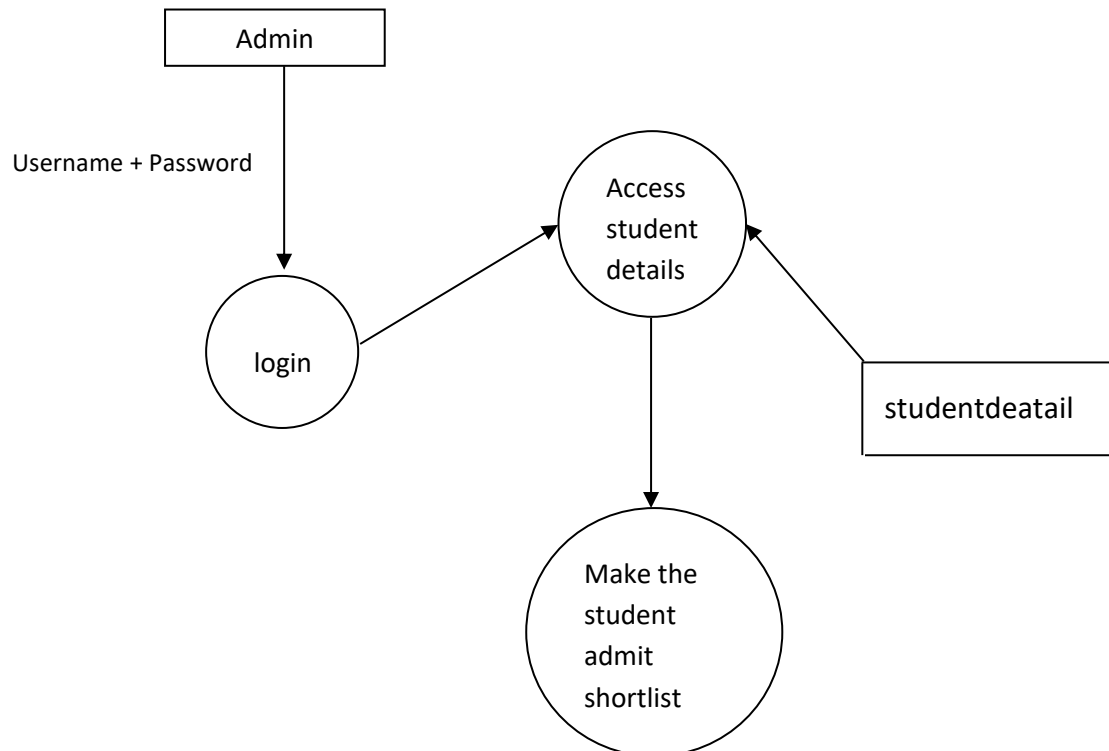
FIELDNAME	DATATYPE
<u>regid</u>	Varchar()
house	Varchar()
post	Varchar()
street	Varchar()
city	Varchar()
pin	Integer
Email-id	Varchar()

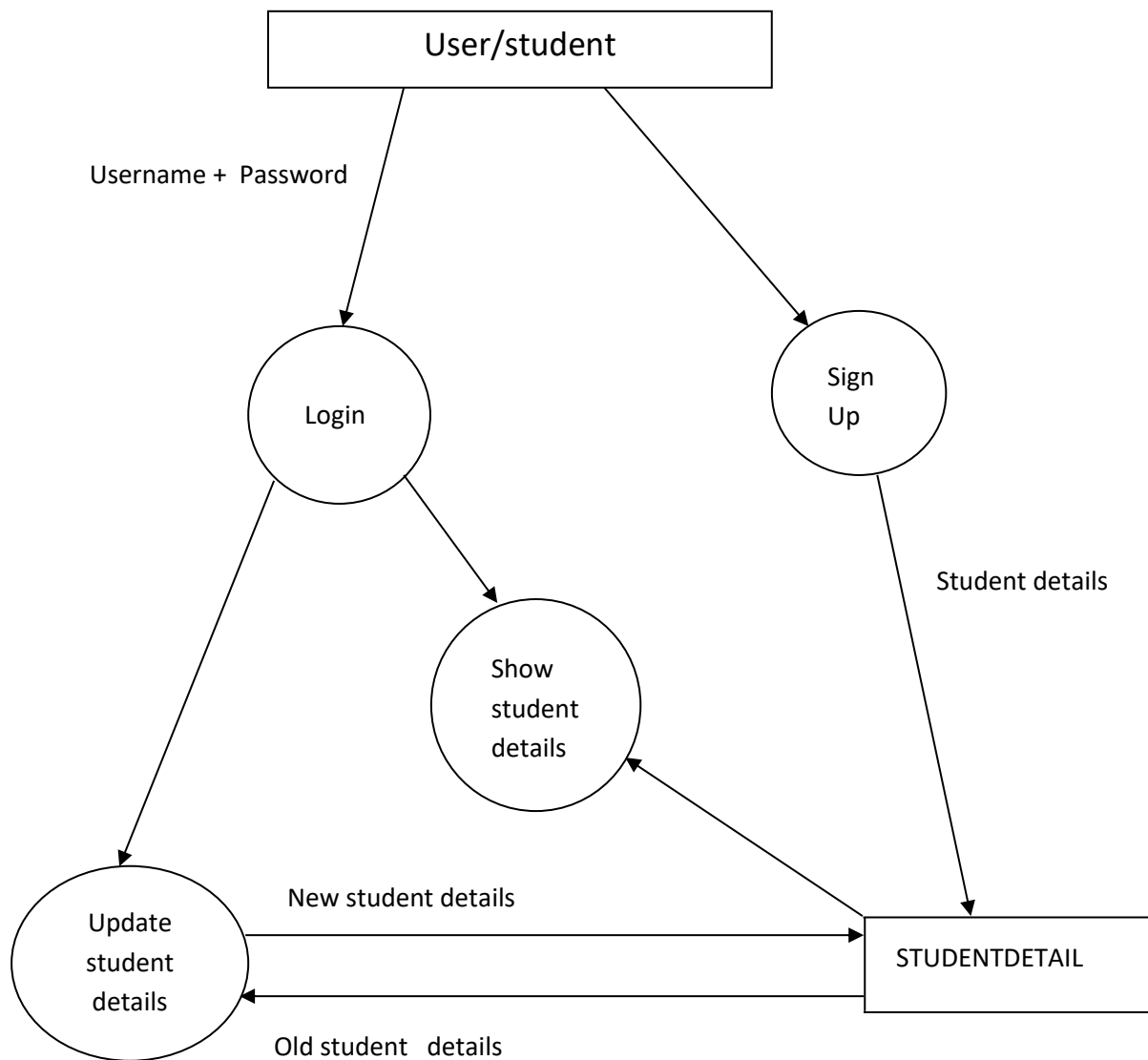
6.4.3 DATAFLOW DIAGRAMS

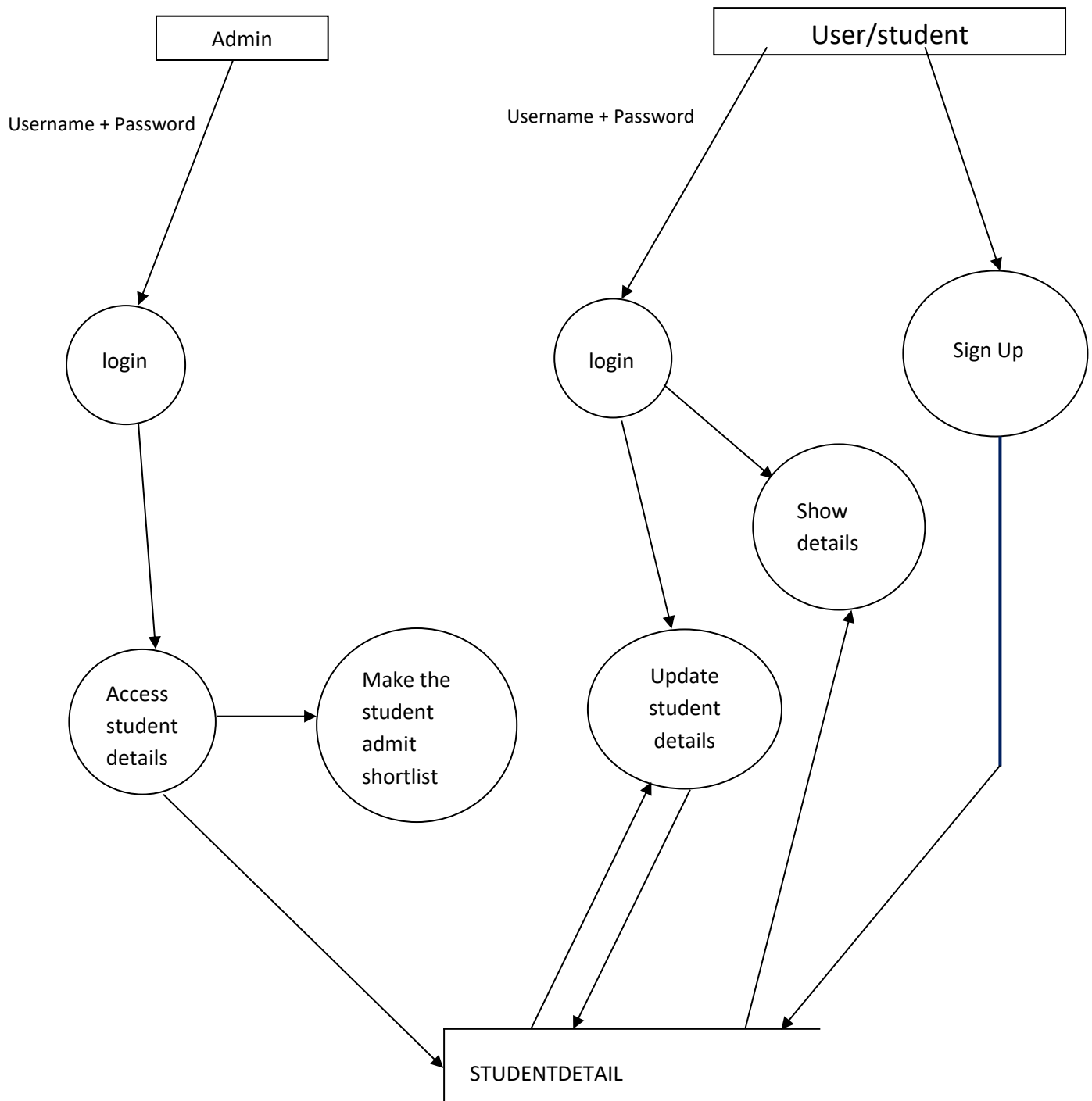
Level 0



Level 1



Level 2

Level 3

7. CODING

A code is an ordered collection symbols to provide unique identification of data. Codes can be used by people who do not with data processing; the following are characters of a good code generation. Characteristics of good coding are

- Uniqueness
- Meaningfulness
- Stability
- Uniform Size and Format
- Simplicity
- Conciseness
- Versatility

The goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and that performs the computation specified by the design. The coding phase affects both testing and maintenance profoundly. As we saw earlier, the time spent in coding is a small percentage of the total software cost , while testing and maintenance consume the major percentage. Thus it should be clear the goal during coding should not be to reduce the implementation cost, but the goal should be to reduce the cost of later phases, even if it means that the cost of this phase has to increase. In other words, the goal during this phase is not to simplify the job of the programmer. Rather the goal should be to simplify the job of the tester and the maintainer.

7.1 CODE EFFICIENCY

7.1.1CORRECTNESS

The tester and the maintainer are not the programmers itself. Therefore it is very important to produce correct and simple code. The code for the proposed system is correct and simple. Anyone who knows HTML and PHP can understand the code very easily.

7.1.2 RELIABILITY

The program must be reliable, that is, it must be execute reliably in a variety of systems. The code for proposed system is consistent.

7.1.3 ROBUSTNESS

The robustness of the code is associated with memory management and exception handling. Since he proposed system is developed with HTML this feature is not ensured profoundly.

7.1.4 DEVELOPMENT EFFICIENCY

The PHP ensures development efficiency by providing compatibility with almost all web servers used today. Extending PHP is pretty easy. It is a full-fledged development environment consisting of a several megabytes of source code. PHP scripts are browser neutral, the script cannot be captured via a eb browser's 'View Source' capabilities.

7.1.5 EXECUTION EFFICIENCY

PHP is easy to learn and runs efficiently and quickly on any compatible web server. The PHP is a parsed language. The parser which execute PHP produce the desired result as HTML code. The code written for the proposed system consists of many PHP files which execute on the web server and return output as HTML to the browser.

7.2 TECHNOLOGY SPECIFICATIONS

7.2.1 PHP (Hyper text Preprocessor)

PHP, which stands for "PHP: Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.

Security

PHP is a powerful language and the interpreter, whether included in a web server as a module or executed as a separate CGI binary, is able to access files, execute commands and open network connections on the server. These properties make anything run on a web server insecure by default. PHP is designed specifically to be a more secure language for writing CGI programs than Perl or C, and with correct selection of compile-time and runtime configuration options, and proper coding practices, it can give you exactly the combination of freedom and security you need.

As there are many different ways of utilizing PHP, there are many configuration options controlling its behavior. A large selection of options guarantees you can use PHP for a lot of purposes, but it also means there are combinations of these options and server configurations that result in an insecure setup.

The configuration flexibility of PHP is equally rivaled by the code flexibility. PHP can be used to build complete server applications, with all the power of a shell user, or it can be used for simple server-side includes with little risk in a tightly controlled environment. How you build that environment, and how secure it is, is largely up to the PHP developer.

Features

PHP manages freeing all resources. Users does not required to free file handle resource, database resources, memory, etc, unless programmer need to free resource during script execution. All resources are released after script execution PHP4 also have reference count feature. For example, memory for variables is shared when it assigned to other variable. If contents have been changed, PHP4 allocate new memory for it. Programmer does not have to use pass by reference for large parameters for better performance with PHP4. It would be a nice section for new PHP users, if there is "Resource Handling" section or like. Explanation about reference count feature in PHP4 would be very helpful to write better PHP4 scripts also.

7.2.2 Macromedia Dream Weaver

This is a web-designing tool from Macromedia. Interactive animated web pages can be created quickly with Macromedia Dream Weaver.

7.2.3 My SQL

My SQL query () sends a unique query (multiple queries are not supported) to the currently active database on the server that's associated with the specified *link_identifier*. The query string should not end with a semicolon. If the link identifier is not specified, the last link opened by My SQL connection is assumed. If no such link is found, it will try to create one as if mysql_connect () was called with no arguments. If by chance no connection is found or established, an E_WARNING level warning is generated.

7.3 OPTIMIZATION OF CODE

Code optimization aims at improving execution efficiency of a program. This is achieved in two ways:

- Redundancies in a program are eliminated.
- Computations in a program are rearranged or rewritten to make it execute efficiently.
- The optimization must not change the meaning of a program.

The ‘Online Admission system ‘ optimizes the code by using the optimization techniques such as dead code elimination and frequency reduction. Thus improves the execution efficiency.

7.4 VALIDATION CHECKS

Validation means observing the behavior of the system. The verification and validation means that will ensure that the output of a phase is consistent with its input and that the output of the phase is consistent with the overall requirements of the system. The ‘College Alumni’ system performed validation by verifying the output of each phase. This is done to ensure that it is consistent with the required output. If not we apply certain mechanisms for repairing and thereby achieved the requirement.

8. IMPLEMENTATION AND MAINTENANCE

Implementation is an activity that is contained throughout the development phase. It is a process of bringing a developed system into operational use and turning it over to the user. The new system and its components are to be tested in a structured and planned manner. A successful system should be delivered and users should have confidence that the system would work efficiently and effectively. The more complex the system being implemented the more involved will be the system analysis and design effort required for implementation.

8.1 IMPLEMENTATION PLANS

The major activities in implementation plan are cost estimation, schedule and milestone determination, project staffing, quality control plans, and controlling and monitoring plans. The implementation plan involves the following:

- Testing to confirm effectiveness.
- Detection and correction of errors.

The system has two modules. They are

- User
- Administrator

Students logging is to apply for the course by filling an application form provided by online. College principal/administrator logging in may also access/search information put up by the students.

8.2 CHANGEOVER METHOD

If the implementation is changed from one data structure to another, the internals of the object need to be changed, that is the data definitions and the implementation of the operations. From the outside, the directory object can continue to be used in the same manner as before, because its interface is not changed.

8.3 TESTING

Software testing is a critical element of the software development cycle. The testing is essential for ensuring the Quality of the software developed and represents the ultimate view of specification, design and code generation. Software testing is defined as the process by

which one detects the defects in the software. Testing is a set of activities that work towards the integration of entire computer based system.

A good test case is one that has a high probability of finding an as-yet undiscovered error. A successful test is one such uncovers or finds such errors. If testing is conducted successfully, it will uncover errors in the software. It also demonstrates that software functions are being performed according to specifications and also behavioral and performance requirements are satisfied. For this, test plans have to be prepared. The implementation of a computer system requires that test data has to be prepared and that all the elements in the system are tested in a planned and efficient manner. Nothing is complete without testing, as it is vital success of the system.

8.3.1 TESTING OBJECTIVES

There are several rules that can serve as testing objectives. They are:

- Testing is process of executing a program and finding a bug.

A good test case is one that has a high probability of finding an undiscovered.

- A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to be working according to the specification, that performance requirements appear to have been met.

8.3.2 UNIT TESTING

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic.

This is the first level of testing. In this, codes are written such that from one module, we can move on to the next module according to the choice we enter.

8.3.3 SYSTEM TESTING

In this, the entire system was tested as a whole with all forms, code, modules and class modules. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. It is a series of different tests that verifies that all system elements have been properly integrated and perform allocated functions. System testing makes logical assumptions that if all parts of the system are correct, the goal will be successfully achieved. Testing is the process of executing the program with the intent of finding errors. Testing cannot show the absence of defects, it can only show that software errors are present.

8.3.4 INTEGRATION TESTING

This testing strategies combines all the modules involved in the system. After the independent modules are tested, dependent modules that use the independent modules are tested. This sequence of testing layers of dependent modules continues until the entire system is constructed.

Though each module individually, they should work after linking them together. Data may be lost across interface and one module can have adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time, conducting test to uncover errors associated with the interface. In the testing the programs are constructed and tested in the small segments.

8.3.5 WHITE BOX TESTING

White-box testing is concerned with testing the implementation of the program. The intent of this testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program. White box testing is also called **structural testing**.

To test the structure of a program, structural testing aims to achieve test cases that will force the desired coverage of different structures. Various criteria have been proposed for this.

There are three different approaches to structural testing: control flow-based testing, data flow-based testing, and mutation testing.

8.3.6 BLACK BOX TESTING

In black-box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or specifications of the program or module, and the internals of the module or the program are not considered for selection of test cases.

In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. This form of testing is also called functional or behavioural testing.

The most obvious functional testing procedure is exhaustive testing. One criterion for generating test cases is to generate them randomly. There are no formal rules for designing test cases for functional testing. In fact, there are no precise criteria for selecting test cases.

8.4 SYSTEM MAINTENANCE

Maintenance involves the software industry captive, typing up the system resources. It means restoring something to its original condition. 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 re-engineered or deployed to another platform. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software

9. SYSTEM SECURITY MEASURES

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files.

9.1 TECHNICAL SECURITY MEASURES

This is associated with technical problems such as exception conditions detected during execution. This exception should be programmed in the system itself. System errors are some operation during the execution may cause it to fail. This is because of logical programming error. So it is essential to handle these problems effectively.

9.2 INFRASTRUCTURAL SECURITY MEASURES

This is associated with computer failure, system crash, disk failure. Physical problems and catastrophes refers to an endless list of problems that includes power air conditioning failure, fire, theft, damage, overwriting disk or tape by mistake.

9.3 ORGANIZATIONAL SECURITY MEASURES

Some problems occur in organization due to the concurrent execution of system. Prevention of labour turnover is another security measure related to the organization.

9.4 PERSONAL RELATED SECURITY MEASURE

This security is concerned with offering security to the persons who are developing and using the system. This is offered by providing username and password for each and every person using the system.

10. COST ESTIMATION OF THE PROJECT

The most cost estimation procedures focus on estimating effort in terms of person-months (PM). By properly including the "overheads" (i.e., the cost of hardware, software, office space, etc.) in the cost of a person-month, effort estimates can be converted into cost. For a software development project, effort and schedule estimates are essential prerequisites for managing the project.

10.1 PERT (PROGRAM EVALUATION AND REVIEW TECHNIQUE)

PERT is a network planning method for managing and controlling large one-time projects. It is a technique for scheduling complicated projects comprising many activities, some of which are interdependent.

A **PERT network** is a flowchart like diagram that depicts the sequence of activities needed to complete a project and the time or costs associated with each activity.

1. All of the major activities in the project are specified.
2. The sequences of these activities are determined
3. A **network diagram** a graphic depiction of the interrelationships among activities, is constructed.
 - a. An **activity** is a work component to be accomplished, and is represented by an arrow on the network diagram.
 - b. An **event (or node)** represents a single point in time that is the beginning or the ending of an activity.
4. Three time estimates for each activity are determined and an expected time is calculated for each activity.
5. The **critical path** is the path of activities and events in the network that will take the longest time to complete
 - a. Delays on any activities on the critical path mean that the project will be delayed.
 - b. **Slack** is the degree of latitude about when various activities can be started without endangering the completion date of the entire project.
6. After the project has begun, actual times for completion of each activity are collected and recorded on the PERT network so that any rescheduling and adjustments can be made as quickly as possible.

10.2 GANTT CHART

A timeline chart can be developed for the entire project. Alternatively, separate charts can be developed for each project function or for each individual working on the project. When multiple bars occur at the same time on the calendar, task concurrency is implied. The diamonds indicate milestones.

Once the information necessary for the generation of a timeline chart has been input, the majority of software project scheduling tools produce *project tables*—a tabular listing of all project tasks, their planned and actual start- and end-dates, and a variety of related information (Figure 1). Used in conjunction with the timeline chart, project tables enable the project manager to track progress.

Phases	20/12/10 to 05/01/11	06/01/11 to 21/01/11	26/01/11 to 07/02/11	11/02/11 to 21/02/11	27/02/11 to 26/04/11	02/05/11 to 29/05/11
feasibility study	✓					
Requirement analysis		✓				
System design			✓			
Detailed design				✓		
Coding and testing					✓	
Operation and maintenance						✓

Figure1

9. FUTURE SCOPE

The future scope of this project is very broad

Few of them are:

- This can be implemented in less time for proper admission process
- This can be accessed anytime anywhere, since it is a web application provided only an internet connection.
- The user had not need to travel a long distance for the admission and his/her time is also saved as a result of this automated system

10. CONCLUSION

This system, being the first We have created in PHP, has proven more difficult than originally imagined. While it may sound simple to fill out a few forms and process the information, much more is involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself.

Debugging can sometimes be a relatively straight forward process, or rather finding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first.

Language used must be simple and easy to understand and compatibility is paramount. If this system were not designed as an entirely web based application, it would not have been possible to recreate its current state of portability.

Overall, the system performs well, and while it does not include all of the features that

may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

11. BIBLIOGRAPHY

- ✓ An Integrated approach to Software Engineering, Narosa publishers, Third edition by Pankaj Jalote.
- ✓ The PHP Manual-www.php.net.
- ✓ PHP Manual-www.w3school.com.
- ✓ Secure PHP Development-Building 50 Practical Applications by Mohammed J Kabir, Wiley Publishing, inc.
- ✓ PHP for Absolute Beginners by Jason Lengstorf

14. APPENDICES

SCREENSHOTS