**PROJECT REPORT**

**ON**

**Complaint Management System**

A Project submitted

in partial fulfilment of the requirement

for the degree of

**MASTER OF COMPUTER APPLICATION**

BY

**PARNEETA BAHUGUNA**

(Enrollment no: PV-15010480)

**Under the Supervision of**

**Miss.Tabish Rao**

**GRAPHIC ERA HILL UNIVERSITY, DEHRADUN**

****

**GRAPHIC ERA HILL UNIVERSITY, DEHRADUN**

**JUNE, 2017**

**ACKNOWLEDGEMENT**

Words are often too weak for revealing one’s deep feeling. As the intensity of sentiments deepens, I humble for the words. I feel it a high privilege for me to express my deep sense of gratitude t all those who have helped me during my project work.

I would first of all like to convey my sincere thanks to **Mr. Naveen Garg** (HOD, Department of Computer Application), for providing me the facilities and valuable advices supports and help whenever necessary. I also express my gratitude to my internal guide Reader **Miss Tabish Rao (Assistant Professor)** Department of Computer Application, and all who encouraged, Supported, assisted and motivated me throughout this project.

I express my deepest gratitude to my Project Guide **Mr. Rajeev Bhatia** for providing me his valuable guidance and encouragement during the course of project.

At last, but certainly not least, I would like to thank to all my friends from their inspiration and constant moral support, directly or indirectly, for the betterment of my project.

**DECLARATION**

I hereby declare that the Industrial Training Report entitled **Complaint Management System** is an authentic record of my own work as requirements of 6-months IndustriaL Training during the period from January to June for the award of degree of MCA **Graphic Era Hill University**.

**(Signature of Student)**

(**Parneeta** **Bahuguna**)

# (1001480)

**CERTIFICATE**

Certified that **PARNEETA BAHUGUNA** ( Enrollment no:PV-15010480) has carried out the dissertation work presented in this report entitled **"Complaint Management System"** for the award of master of computer application from **GRAPHIC ERA HILL UNIVERSITY, DEHRADUN** under my supervision.The thesis embodies results of original work, and studies as are carried out by the student himself and the contents of the thesis do not form the basis for the award of any degree to the candidate or to anybody else from this or any other University /Institution.

(Miss Tabish Rao) (Rajiv Bhatia)

Assistant Professor Project Guid

Graphic Era Hill University Excess Computer Education

(Mr. Ishteyaaq Ahmed) External Examiner

Head of Department

Depatment of Computer Application

Graphic Era Hill University,Dehradun

Date:

**CONTENTS AT A GLANCE**

* 1. Introduction 7
  2. Objective 8-9
  3. System Analysis 10-13

i. Preliminary Investigation 14-16

ii. General principles of system 17-18

Investigation

* 1. Feasibility Study 19-22
  2. Software Paradigm used 23-25
  3. Software Process Model 26-28
  4. Software And Hardware Requirement 29-30

Specification

* 1. Software Platform Used 31-33
  2. Information System 34-37
  3. Design& Database Description 38
     1. Category of the Project:DBMS 39-42
     2. Entity Relationship Diagram 43-46
     3. Data Flow Diagram 47-52

11. Input / Output Screen Shorts 53-64

12. Testing and Maintenance 65-70

13. Scope and Limitations 71

14. Bibilography 72

15. Conclusion 73

# INTRODUCTION

For all types of Complaint we need to manage their records in such a way that they can retrieve the information as per requirement. They need a set of rulebooks, such as Types of Complaints, Customers etc. Unless the transactions are recorded properly, Engineer will not be in a position to know where exactly he stands. Therefore, for any business record keeping is of foremost importance.

Handling the Complaints manually is quite difficult so this software provides aid to the customer by providing easy access to the records. This will improve the efficiency of the work because manually he would take a long time to search for and keep track of all the records of the complaints taking place in the company.

Following is the complete **cycle of Complaint Management System:**

1. The Detail ofCustomers and other detail from opening and day-to-day transactions are first recorded in a file.
2. Periodically these transactions are transferred to concerned Database.
3. At the end of every information balanced can be seen in the reports.

### An additional feature of the software is that it provides security to the Managment. It is Password-Protectedsoftware, which provides for three types of logins, which are Engineer Login ,User login and Administrator login.

Another salient feature is the **Reports Generation** facility, which includes the reports for daily transactions and monthly reports. Above-mentioned are some of the prime features of the software developed. It also covers all the basic tasks covered by Information.

## OBJECTIVE

The main objective of the project is to computerize all the aspects related to Complaint Management System.

The main aim is to develop software to get fast solutions of problem from Engineerat user’s choice; also to completely eliminate the errors occurred while resolving the complaints.

In brief, the following points describes the main objectives to be achieved through this software:

* To fully **computerize** all the aspects of Complaint Management System.
* To design a system that gives the **faster** results and is **more efficient.**
* To provide a **user-friendly** system that is easier to access.
* To make the package easy to access by providing **Menu-Driven** options.
* To design a system that is **reliable.**
* System could add new records maintain them in database.
* To develop **highly secure** software system to avoid any unauthorized user to access any kind of information by providing some security features like:

**Administrator: -**Administrator has his own user name and password to access the package. The Administrator has privileges to update, delete and create user. Administrators can also rollback the database, if required.

**User: *-*** User has his own user name and password. User has all the privileges except deletion, updating and creation of the user. User cannot rollback the database without the permission of administrator.

* To maintain integrity and prevents the unauthorized access.
* To provide better information facilities.

The main objective of the system is to improve the Online Complaint System, reduce paperwork, better transaction handling, up to date transaction and possible reduction in staffing requirements and faster retrieval of information and to generate Quick reports. **In short, its main aim is to have a faster and accurate information.**

**SYSTEM ANALYSIS**

#### Identification of Need

Performing required and defined functions manually may result in a number of different Complications/Problems.

**The Problem of Reliability**

However, manual approach of maintaining Complaints information may be reliable but not often so. The reliability of manual approach depends heavily upon the person's nature, behavior, interpersonal relationships etc. Therefore a reliable system is required that can overcome these problems. Since the reliability of computer systems has proven its worth over the time being hence it can be relied upon.

The Reliability means the probability that the software will operate as expected over a specified time interval. The notion of reliability is relative if the consequence of a software error is not serious, the incorrect software may still be reliable.

**The Problem of Accuracy**

A system is inaccurate when processing is error prone. For e.g. In Online Complaint Management System information posting procedure is complex and the number of transactions is large, a fair number of errors may occur. Therefore, routine, transaction based manual procedures are basically suitable for conversion to computer-based methods of processing because the computer is far more accurate than human beings, provided the software is written properly.

**The Problem of Economy**

Existing manual approach of information suffers from the problem of economy.A group of manpower isrequired for transmitting, processing, and storing information that are very costly.

The online computer solution of the problem will eliminate the economy problem.

**The Problem of Timeliness**

The manual approach suffers from the problem of timeliness because information is available but cannot be retrieved when and where it is needed.

The computer solution may provide the fingertip access to information as desired.

###### The Problem of Large Paper Work

###### In the manual approach, end user has to maintain a large amount of files and papers. It is difficult to maintain the files and paper in an organized way for a long period of time. It may get damaged by some means of environmental factors.

Computerized system solves this problem, as backup can be retained with the data.

###### The Problem of Redundancy

It is one of the measure problem faced during manual systems. Redundancy refers to the repetition of the data. Whenever, there is an updating required, it must be updated every time. Redundancy can be removed in the computerized system.***There was need to solve the problems faced. Thus new additional features are incorporated in the system.***

###### Correctness

A program is functionally correct if it behaves according to the specification of the functions it should provide (called functional requirements specifications). Correctness is an absolute quality; any deviation from the requirements makes the systems incorrect, regardless of how minor or serious is the consequences of the deviation.

**Robustness**

A program is robust if it behaves reasonably, even in circumstances that were not anticipated in the requirements specification. Robustness is one of the major requirements of the customer.

###### User Friendliness

A software system is user friendly if its human users find it easy to use. An application that is used by novice programmers qualifies as user friendly by virtue of different properties than an application that is used by expert programmers.

The user interface is an important component of user friendliness. End user wants a system from the developer that decreases his effort to learn, operate, prepare input and interpret output from the system.

Verifiability

A software system is verifiable, if its properties can be verified easily. Modular design, disciplined coding practices, and the use of an appropriate programming language all contribute to verifiability.

Maintainability

The term software maintenance is commonly used to refer to the modifications that are made to a software system after its initial release.

**Reusability**

Reusability refers to the extent to which a program (parse of program) can be reused in another applications - related to the packaging and scope of the function that the program performs. In product evolution, we modify a product to build a new version of that same product; in product reuse, we use it - perhaps with minor changes - to build another product.

**Portability**

System is portable, if it can run in different environments. The term environment can refer to a hardware platform or a software environment. Portability refers to run a system on different hardware platforms.

**PRELIMINARY INVESTIGATION**

The concept of life cycle of a system is central to **System Investigation**. Every system moves through the several phases of a life cycle during its development, after which it functions with only minor maintenance for a period of years. The system gradually deteriorates to the point where it ceases to function effectively, and a new life cycle begins with the development of a new system.

**System Investigation** is the process of finding every aspect of required system. **System Investigation** provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system.

It certainly seems simple enough – ask the customer, the users, and others what the objective for the system or product are, what is to be accomplished, how the system or product fits into the needs of the business, and finally, how the system or product is to be used on a day - to - day basis.

There are number of **problems** during this phase:

The boundary of the system is ill defined or the customers / users specify unnecessary technical details that may confuse, rather than clarify, overall system objectives.

**Problem of Understanding**

The customers / users are not completely sure of what is needed, have a poor understanding of the capabilities and limitations of their computing environment, don’t have a full understanding of the problem, have trouble communicating needs to the system engineer, omit information that is believed to be “obvious”, specify requirements that conflict with the needs of other customer / users, or specify requirements that are ambiguous and untestable.

**Problem of Volatility**

The requirement changes over time. The systems developer built is seldom finished. Demands for new abilities keep coming from user’s side.

To help overcome these problems, system engineers must Approach the requirements gathering activity in an organized manner. The System Investigation is a complex activity with many variables, and there is ample scope to permit wide variety serious blunders.

A **System Investigation** is a project that involves people working together toward the common goal of improving the information system. These problems are due to personality traits that to some degree prevent a system from being properly developed or properly used after development:

**Commitment to the Old Information System**

Users of the existing information system who do not wish to learn to use a new system may oppose it because they are committed to the old system.

**Resistance to Change**

Everyone resists change; to an extent, we are all creatures of habit. Because the usual purpose of a System Investigation is to bring about change, it is not surprising that many people are unwilling to cooperate wholeheartedly with it.

**Embarrassment**

Because people are embarrassed by a lack of knowledge about computer system or by an inability to specify their information needs fully, they may avoid participation in the system investigation.

**Fear of Job Loss**

As employees are well aware, new systems frequently eliminate jobs or substantially alter a job’s contents.

**Lack of Interest among Key Personnel**

Often it is difficult to capture the interest key personnel. Many managers are quite busy and are disinclined to give their attention to a system project; even if the new system will serve them.

**GENERAL PRINCIPLES OF SYSTEM INVESTIGATION**

Several general principles of System Investigation are of paramount importance to the successful completion of a project and are examined briefly below:

**Involve Managers in the Project**

In all system projects, managers of the user organization should be involved, and in major systems projects, senior managers as well as managers at other level in multiple areas of the organization should be involved.

**Involve Users in the Project**

The necessity to involve users in system project appears self-evident, and yet system specialist often attempts to develop new system with no or minimal user participation. A system that is designed without extensive interaction with users usually becomes an unpleasant surprise to them and is not likely to be utilized. Users generally react negatively to any system that has been designed without their involvement.

**Plan the Project within the Context of Long Range System Planning**

Without adequate long range planning for the entire information system, individual projects are unlikely to serve the overall interest of the organization satisfactorily. Each proposed project must be evaluated in terms of the goals of entire organization.

**Establish the Objectives and Scope of a Proposed System Early**

If the objectives and scope of new system are formally established at about the time the project is begun, all effort is focused explicitly on achieving the objectives. Unfocused effort is likely to be frittered away on tangential activities of little value to the organization.

**Users Structured Approach**

System Investigation should use a structured methodology, consisting of a series of steps within each phase, each done more or less in sequence. This provides a discernible structure to the project. This structure also serves to increase the systematic nature of the activity and thereby reduces the extent of wasted effort.

**Establish Priorities for Projects**

There often appears to be an infinite number of possible objectives for a system. The objectives with highest priority should be considered before those with the lower priorities.

### FEASIBILITY STUDY

**Feasibility is the determination of whether or not a project is worth doing**. The process followed in making this determination is called a Feasibility Study. Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report.

#### DIFFERENT TYPES OF FEASIBILITY

In the conduct of the feasibility study, the analyst will usually consider seven distinct, but inter-related types of feasibility. They are:

**Technical Feasibility**

This is concerned with specifying equipment and software that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include:

* The facility to produce outputs in a given time.
* Response time under certain conditions.
* Ability to process a certain volume of transaction at a particular speed.
* Facility to communicate data to distant location*.*

At the feasibility stage, it is desirable that two or three different configurations will be pursued that satisfy the key technical requirements but which represent different levels of ambition and cost. Out of all types of feasibility, technical feasibility generally is the most difficult to determine.

There are a number of technical issues, which are raised during the feasibility stage of investigation. In respect of proposed system we found that:

* Necessary technology exists to do what is suggested
* Proposed equipment has the technical capacity to hold the data required using the new system.
* System is upgradeable if developed.
* There are technical guarantees of accuracy, reliability, ease of use and data security.Thus the system is technically feasible.

**Operational Feasibility**

It is mainly related to human organizational and political aspects. The points to be considered are:

* **What changes will be brought with the system?**
* **What organizational structures are disturbed?**
* **What new skills will be required?**
* **Do the existing staff members have these skills? If not, can they be trained in due course of time?**

People are inherent to change. An estimate should be made of how strong a reaction the user staff is likely to have towards the development of a computerized system. Therefore it is understandable that the introduction of a candidate system requires a special effort to educate, sell, and train the staff on new ways of conducting business.

The user can update their database and also use the processed records.The second such community was the developer. They too were getting benefited as they were getting this software, which can be enhanced very much.

As the proposed system is very user friendly, and asked by the customer showing the willingness to adapt. Thus making the proposed system operational feasible.

**Economic Feasibility**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposal system. More commonly known as Cost / Benefit analysis; the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs.

Among the most important information contained in feasibility study is cost benefit analysis-an assessment of the economic justification for a computer based system project. Cost-benefit analysis delineates costs for project development and weight them against tangible and intangible benefits of a system.

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. Otherwise, further justification or alterations in the proposed system will have to be made if it is to have a chance for being approved

**Social Feasibility**

Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project being accepted by the group directly affected by the proposed system change. This project is proved to be socially acceptable and thus social feasible.

**Management Feasibility**

It is a determination of whether a proposed project will be acceptable to management. This is also accepted by the management therefore Management Feasibility is present here.

**Legal Feasibility**

Legal feasibility is a determination of whether a proposed project infringes on known Acts, Statues, as well as any pending legislation.

**Time Feasibility**

Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame.

**SOFTWARE ENGINEERING PARADIGM USED**

To solve actual problems, developer must incorporate a development strategy that encompasses the process, methods, and tools. This strategy is often referred to as a process model or a software engineering paradigm. Whenever software is engineered, a paradigm is used to design it. The “**Fourth Generation Techniques**” is the paradigm used in this development.

The term **Fourth Generation Techniques** encompasses a broad array of software tools that have one thing in common; each enables the software engineer to specify some characteristics of software at a high level. The tool then automatically generates source code based on the developer’s specification. There is little debate that the higher the level at which software can be specified to a machine, the faster a program can be built. The **4GT** paradigm for software engineering focuses on the ability to specify software using specialized language forms or a graphic notation that describes the problem to be solved in terms that the customer can understand.

Currently software development environment that supports the 4GT paradigm includes some or all of the following tools: **Non-Procedural Languages for database query, Report Generation, Data Manipulation, Screen Interaction and Definition, Code Generation; High-Level Graphics Capability; Spreadsheet Capability and Automated Generation of HTML and similar languages used for Website creation using advanced software tools**. Initially many of the tools noted previously were available only for very specific application domains, but today 4GT environments have been extended to address most software application categories.

Like other paradigms, 4GT begins with a requirement-gathering step. Ideally, the customer would describe requirements and these would be directly translated into an operational prototype. But this is unworkable. The customer may be unsure of what is required, may be ambiguous in specifying facts that are known, and may be unable or unwilling to specify information in a manner that a 4GT tool can consume. For this reason, the customer / developer dialog described for other process models remains an essential part of the 4GT approach.

For small applications, it may be possible to move directly from requirement gathering step to implementation using a non-procedural fourth generation language (4GL) or a model composed of a network of graphical icons. However, for larger efforts, it is necessary to develop a design strategy for the system, even if a 4GL is to be used. The use of 4GT without design (for larger projects) will cause the same difficulties (poor quality, poor maintainability, poor customer acceptance) that have been encountered when developing software using conventional approaches.

Like all software engineering paradigms, the 4GT model has its own advantages and disadvantages. Proponents claim dramatic reduction in software development time and greatly improved productivity for people who build software. Opponent claim that current 4GT tools are not all that much easier to use the programming languages, that the resultant source code produced by such tools is inefficient and that the maintainability of large software systems developed using 4GT is open to the question.

There is some merit in the claims of both sides and it is possible to summarize the current the state of 4GT approaches: -

1. The use of 4GT is viable approach for many different application areas.

1. Data collected from companies that use 4GT indicate that the time required to produce software is greatly reduced for small intermediate applications and that the amount of design and analysis for small applications is also reduced.
2. However the use of 4GT for large software development efforts demands as much or more analysis, design and testing (Software Engineering Activities) to achieve substantial time savings that result from the elimination of coding.

**Here in this project**, the paradigm used is **Fourth Generation Techniques (4GT)**. As the software tools that are being used in this system are **‘Java’** and **‘File Handling’**, both are 4 Generation Tools.

As the project neither belongs to large application category nor to smaller one, hence one can’t discard the requirement-gathering step completely. The project needs the intermediate efforts. The requirement gathering for the successful completion of project was performed in Analysis Phase

# 

# SOFTWARE PROCESS MODELS

* **The waterfall model**
  + Plan-driven model. Separate and distinct phases of specification and development.
* **Incremental development**
  + Specification, development and validation are interleaved. May be plan-driven or agile.

**Reuse-oriented software engineering**

* + The system is assembled from existing components. May be plan-driven or agile.

In practice, most large systems are developed using a process that incorporates elements from all of these models.

**The waterfall model**

2.1.Waterfall-model.eps

**Waterfall model phases**

* There are separate identified phases in the waterfall model:
  + Requirements analysis and definition
  + System and software design
  + Implementation and unit testing
  + Integration and system testing
  + Operation and maintenance
* The main drawback of the waterfall model is the difficulty of accommodating change after the process is underway. In principle, a phase has to be complete before moving onto the next phase.

**Waterfall model problems**

* Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
* Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
* Few business systems have stable requirements.
* The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.

**In those circumstances, the plan-driven nature of the waterfall model helps coordinate the work.**

**SOFTWARE AND HARDWARE REQUIREMENTS**

The technical needs of a user that are required for implementation of the new system are as follows.

**HARDWARE RESOURCES:**

1. **WINDOWS 2000 Server: (Standard Configuration)**

* Pentium IV based CPU
* 512 MB RAM
* 40 GB HDD
* 15'' color monitor
* Scroll mouse
* 104 keyboard
* VRAM with SVGA card

1. **Printer [If required]**

**SOFTWARE RESOURCES**

OPERATING SYSTEM : WINDOWS

* + WAMP SERVER
  + XAMPP SERVER

FRONTEND : PHP

LANGUAGE : CSS,QUERY,JAVASCRIPT BACKEND : MY SQL

**Performance Factor**

The system's performance heavily depends upon hardware configuration. Using latest hardware configuration could rectify hardware problems.

Application on which the system is being developed is the next factor that might affect the performance of the whole system. An application build on GUI is not as fast as an application build on 3GLs but on account of its compatibility to any RDBMS package and its user friendliness makes it a popular application development tool. Since ours package is based on RDBMS hence File Handling as backend and Java as front-end tool has been used extensively.

# SOFTWARE PLATFORM USE

**ABOUT PHP**

PHP is a server site scripting language designed primarily for web development but also used for web development but also used for general purpose programming language. Originally created by RasmusLerdorf in 1994,the PHP reference implementation is now produced by the PHP Development Team. PHP originally stood for the recursive acronyms PHP:Hypertext Preprocessor.

PHP code may be embedde into HTML and HTML5 markup,or it can be used in combination with various web template systems,web content management system and web frameworks.PHP code is usually processed by a PHP Interpreter implemented as a module in web server or as a Common Gateway Interface(GUI) executable.

Scripts written in PHP executives faster than those written in other scripting language, with numerous independent benchmarks putting the language ahead

Of competing alternatives like JSP,ASP.NET and PERL. The PHP 5.0 engine was completely redesigned with an optimized memory manager to improve performance, and is noticeable faster than previous version.

* Portability
* Ease of Use
* Open Source
* Performance

# ABOUT HTML

#### Hypertext Markup Language is the standard markup language for creating web pages and web applications.With CSS and JavaScript it forms a triad of cornerstone technologies for the world wide web.Web Browsers HTML documents from a webserver or from local storage and render them into multimedia web pages.

#### Describes the structures of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. HTML constructs,images and other objects ,such as Interactive forms, mqy be embedded into a rendered page. It provides a means to create structured documents.

#### By denoting structural semantics for text such as headings ,paragraphs,lists ,links ,quotes and other items.HTMLElements are delineated by tags,written using angle brackets. Tags such as<img/> and <input/>introduce content into the page directly.

#### Others such as <p>…</p>surround and provide information about document text& may include other tags as sub-elements. Browsers do not display the HTML tags,but use them to interpret the content of the page.HTML can embedded programs written in a scripting language such as JavaScript which affect the behavior and content. The world wide web consurtiom(W3C)maintainer of both the HTML and the CSS standards , has encouraged the use of CSS over explicit presentational HTML since 1997.

#### ABOUT CSS

#### Cascading stylesheet is a stylesheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces HTML and XHTML,the language can be applied to any XML document . Along with HTML and JavaScript ,CSS is a cornerstone technology used by most website to create visually engaging webpages, user interface for web applications, and user interface for many mobile applications.CSS is designed primarily to enable the separation of presentation and content ,including aspects such as the layout, colors and fonts. This separation can improve content accessability,provide more flexibility and control in the specification of presentation characteristics,enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

#### ABOUT jQuery

#### jQuery is a cross platform JavaScript library designed to simplify the client side scripting of HTML.It is free open source software using using the permissive MIT licence. Web analysis indicates that it is the most widely deployed JavaScrip library by a large margin.

**About MySQL**

**MySQL** is an open-source relational **database** management system(RDBMS) based on Structured Query Language(SQL). Mysql runs on virtually all platforms, including Linux, UNIX and WINDOWS. It is a central of the LAMP open-source web application software stack(and other "AMP "stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". It has become the leading database choice for web-based properties including Facebook , Twitter, YouTube, Yahoo! and many more.TheMysql server software itself and the client libraries use dual-licensing distribution.Mysql is a fast, easy-to-use RDBMS being used for many small and big bussiness. It is very friendly to PHP, the most appreciated language foe web development.

# INFORMATION SYSTEM

### MEANING AND OBJECTIVES

Despite the fact that the computer is nothing more than a tool for processing data, many managers view it as the central element in an information system. This attitude tends to overrate and distort the role of computer. Its real role is to provide information for decisions and for planning a controlling operations.

Judging from the business press, the brave new world of management information system (MIS) is upon us. These is hardly a business magazine today that does not contain articles on information systems, data banks and related subjects. Despite this proliferation of books, articles, seminars and courses surrounding this area, few efforts have managed to synthesize the separate subjects of management, information, and systems and to show how these are related to computers.

**A management Information system Defined**

MIS is not new; only its computerization is new. Before computers, MIS techniques existed to supply managers with the information that would permit them to plan and control operations. The computer has added one or more dimensions, such as speed, accuracy, and increased volumes of data, that permit the consideration of more alternatives in a decision.

The scope and purpose of MIS is better understood if each part of them is defined.

**Management**

Management has been defined in a variety of ways, but for our purposes it comprises the processes or activities that describe what managers do in the operation of their organization: Plan, organize, initiate, and control operations. They plan by setting strategies and goals and selecting the

best course of action to achieve the plan. They organize thetask necessary for the operational plan, set these tasks up into homogeneous groups, and assign authority delegation, they control the performance of the work by setting performance standards and avoiding deviations from standard.

Because decision making is such a fundamental prerequisite to each of the foregoing processes, the job of an MIS becomes of facilitating decisions necessary for planning, organizing, and controlling the work and functions of the business.

**Information**

Data must be distinguished from information, and this distinction is clear and important for our purposes. Data are facts and figures that are not currently being used in a decision process and usually take the form of historical records that are recorded and filed without immediate intent to retrieve for decision making. An example would be and one of the supporting documents, ledgers, and so on that comprise the source material for profit and loss statements. Such material would only be of historical interest to an external auditor.

Information consists of data that have been retrieved, processed, or other- wise used for informative or inference purposes, argument or as a basis for supporting documents already mentioned, but in this case the data could be used by an internal auditor, the management service department of an external auditor, or internal management for profit planning and control or for other decision-making purposes.

**Systems**

A system can be described simply as a set of elements joined together for a common objective. A subsystem is part of a larger system with which we are conceded. All systems are parts of larger systems. For our purposes the organization isthe system, and the parts (divisions, departments, functions, units, etc.) are the subsystems.

Whereas we have achieved a very high degree of degree of automation and joining together of subsystems in scientific, mechanical, and factory manufacturing operations, we have barely scratched the surface of applying systems principles to organizational or business systems, The concept of synergism has not generally been applied to the business organization, particularly as it applies to the integration of the subsystems through information interchange. Marketing, operations, and finance are frequently on diverse parts and working at cross purposes. The systems concept of MIS is therefore on of optimizing the output of the organization by connecting the operating subsystems through the medium of information exchange.

A business comes into existence primarily to make profit. In this attempt it provides goods and services to the community. It also provides means of livelihood to some members of the community by creating employment opportunities. These are some subsidiary or secondary motives. The all-important motive of a business, however, is to make profit. **“Profit to a business is like food to a human body: the body must grow and develop - with the assistance of food. Take away the supply of food and the body wastes away and dies.”** Every businessman, therefore, is interested to find out the amount of profit earned (or loss incurred) during a certain period (called‘accountingperiod’), usually a year. In order to find out the amount of profit or loss, he must have a complete and systematic record of the business transactions (i.e., dealings in money or money’s worth) entered into during the year.

Accounting helps a business in having a complete and systematic record of its business transactions in terms of money, reporting results of business activities and interpreting such results for purposes of effective control of future operations or activities. In view of this helpful role of

accounting, it has often been called the **language of business**. Like any language, it can never express our thoughts with absolute precision and clarity. Our task of learning this language is complicated by the fact that many of the terms used in accounting mean almost, **but not quite**, the same as they mean in everyday life. You must learn to think of these terms in their accounting rather than their popular meaning.

#### Advantages of Information System

The following are the advantages of Information System:

1. **Complete record:**It helps in having a complete and systematic record of Students.
2. **Information regarding progress and position:**It gives information about the students made by the teacher at the close of the period and its final condition. The basic function of information is to supply meaningful information about the class activities of the Hospital/Nursing Home to the owners and the managers.
3. **Assistance to management:**It assists the management in
   1. Planning the activities of the Hospital/Nursing Home in future,
   2. Taking decisions in certain matter, and controlling the Hospital/Nursing Home activities according to the schedules of time and production targets.
4. **Comparative study:**It facilitates comparative study of current year’s status, etc., with those of the previous years.
5. **Compliance with legal formalities:**It helps in complying with certain legal formalities, like filing of income tax and sale-tax returns.

# 

# 

# DESIGN & DESCRIPTION

# Architectural Design

Architectural Design represents the structure of data and program components that are required to build this computer based system.

The software requirement can be mapped into various representation of the design model. Structure design is often characterized often as a dataflow oriented design method because it provides a convenient transition from a DFD to a software.

Data Design of Accounts Managing Package creates a model of data and information that is represented at a high level of abstraction.

Architectural design is represented through **ERDs, DFDs, Data Structure used.**

* **Interface Design:**

Once task analysis has been completed, all tasks required by the end user have been identified in details and interface design activity commences.

Firstly in the design phase the goals are identified which, include a consideration of the usefulness of the task, its effectiveness in accomplishing the overriding official objectives and the degree to which the task can be learn quickly with the ultimate implementation of the task and also the intention.

Each goal and intention must be mapped in to a sequence of specific actions.

**Interface design includes form designs and layouts.**

* **Component-level Design**

This design is also being called procedural design, the intent is to translate the design model into operational software. The main construct of this design is sequence, condition and repetition.

**Pseudo codes and Flow Charts are used to represent the component level design through these entire three constructs*.***

**Admin**

|  |  |
| --- | --- |
| Field Name | Data Type |
| Sid | AutoNumber |
| Sname | Text |
| Spass | Text |
| Address | Text |
| Email | Text |
| Smobile | Number |
| Datetime | Date/Time |

**Complaints**

|  |  |
| --- | --- |
| Field Name | Data Type |
| Compid | AutoNumber |
| cid | Number |
| Cname | Text |
| Comptype | Text |
| Comptitle | Text |
| Compdesc | Text |
| Compstatus | Text |
| Eid | Number |
| Ename | Text |
| Ecomm | Text |
| Compdate | Date/Time |
| Compclosedate | Date/Time |

**Customer**

|  |  |
| --- | --- |
| Field Name | Data Type |
| Cid | Number |
| Cname | Text |
| Cpass | Text |
| Caddress | Text |
| Cemail | Text |
| Cmobile | Number |
| Datetime | Date/Time |

**Engineer**

|  |  |
| --- | --- |
| Field Name | Data Type |
| Eid | AutoNumber |
| Ename | Text |
| Epass | Text |
| Address | Text |
| Email | Text |
| Emobile | Number |
| Datetime | Date/Time |

**ENTITY RELATIONSHIP DIAGRAM**

Password

User Name

Administrator

Password

User Name

Engineer

User ID

Password

User

**ADMINISTRATOR**

Administrator

**Engineer**

Engineer

**User**

Customer

Complains

**DFD**

**(Data Flow Diagram)**

**Level 0**

Username

**Admin**

**Option**

Password Login Admin

**Admin**

Not Validate

**Engineer**

**Option**

Username  **Login**

**Engineer**

Password Login Engineer

Not Validate

Username

Password Login User

**User Registration**

**User**

Not Validate

**Level 1**

**Admin DFD:**

**ername**

**Login Admin (1)**

**Not validate**

**Admin**

**Password**

**Engineer DFD:**

**Username**

**Not validate**

**Engineer**

**Password**

**Login Engineer**

**User DFD:**

**rname**

**Password**

**Not validate**

**User**

**Login User**

**Data Flow Diagrams**

The data flow diagram is one of the most improvement tools used by the system analyst DeMacro (1978) NadGandSarson (1979) popularized the use if the data flow diagram as modeling tools through their structured system analysis methodologies.

A data flow diagram should be the first tool used by system analyst to model system components. These components are the system processes; the data used by this processes and external entities that interact with the system and the information flows in the system.

There are four kinds of system components:

**1. Process:** Process show what system does. Each process has one or more data inputs and produces one or more data output, Circles in a data flow diagram represent process. Each process has unique name and number. This name and number appear inside the circle that represents the processes in a data flow diagram.

**2. Data Stores:** File or data store is depositary of data. They contain data that is retained in the system. Processes can enter the data into a data store or retrieve data from the data store. Each data store is represented by thin line in the data flow diagram and each data store has a unique name.

**3. External Entities:** External entities are outside the system but they either supply input data into the system or use the system output, they are entities which the designer has no control. Square or rectangle may represent external entities that supply data into a system or sometimes called sources. External entities that use the system data are sometimes called sinks.

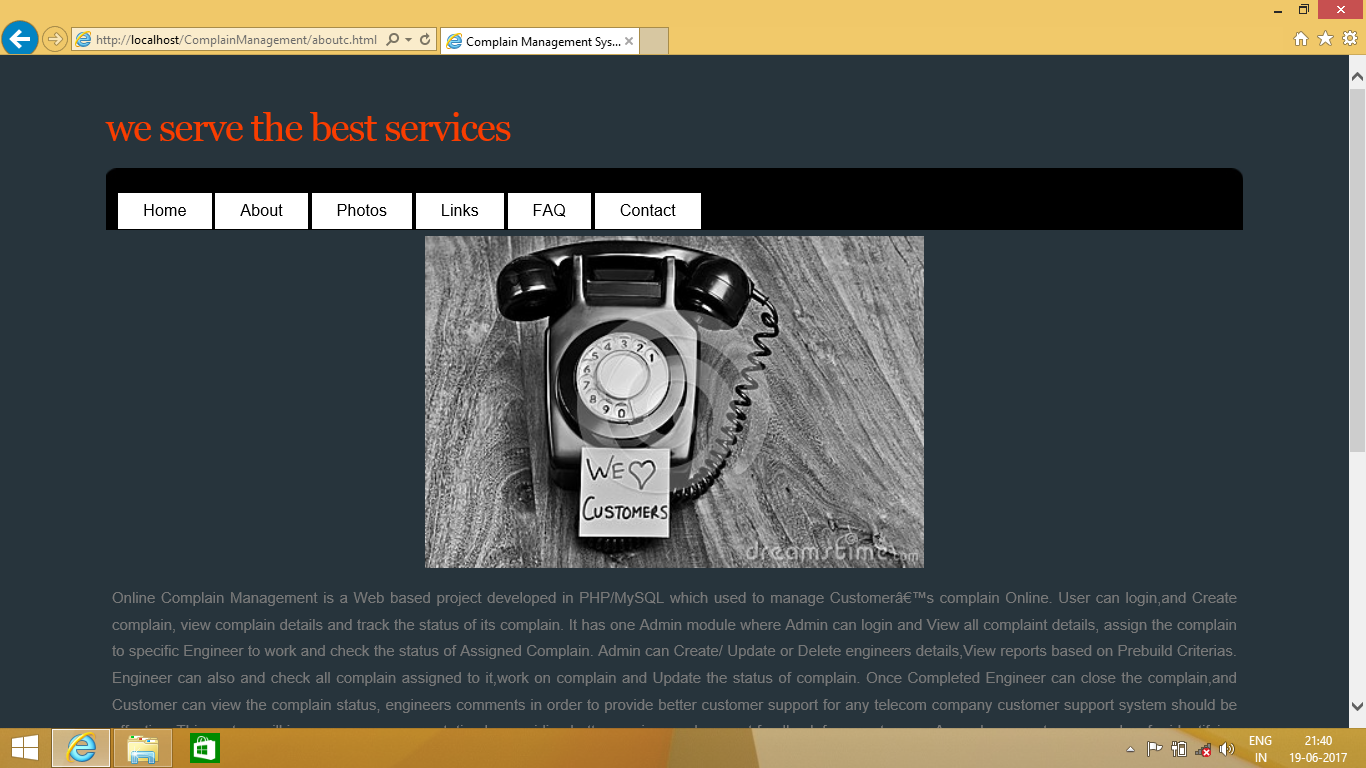
**4. Data Flows:** Dataflow model the passage of data in the system and are represented lines joining system components. An arrow indicates the direction of the flow and the line labeled by the name of the data flow.

**5. Storage:**It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

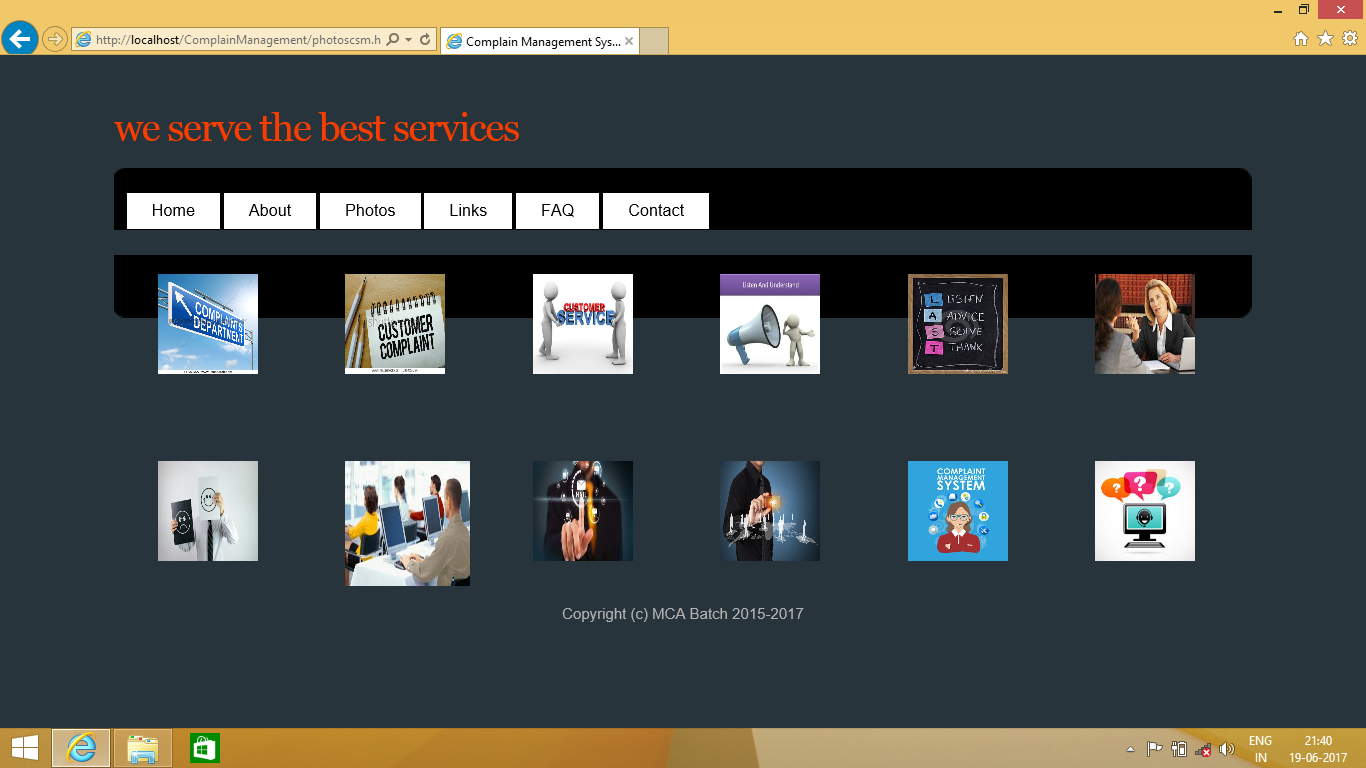
**Input / Output Screen Short:**

****

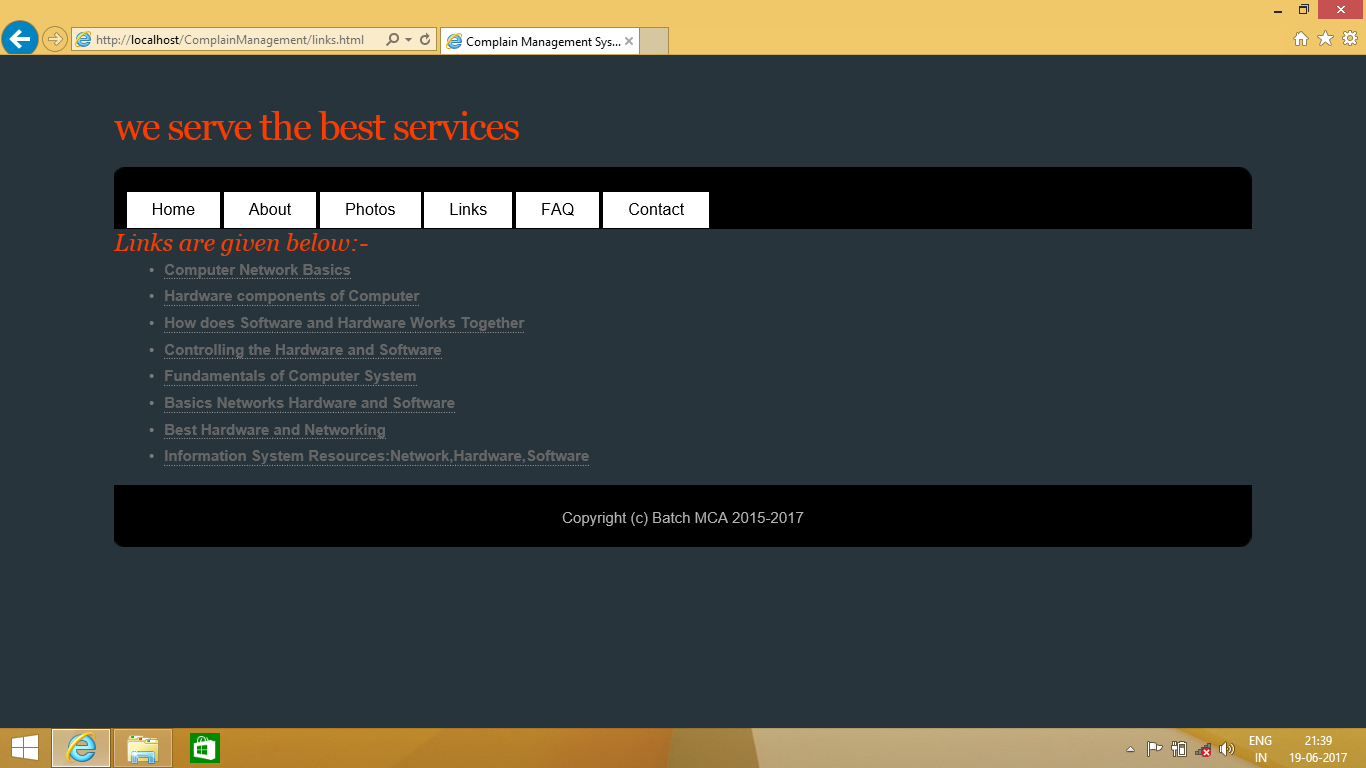
**Fig 1: HomePage**

****

**Fig 2: About**

****

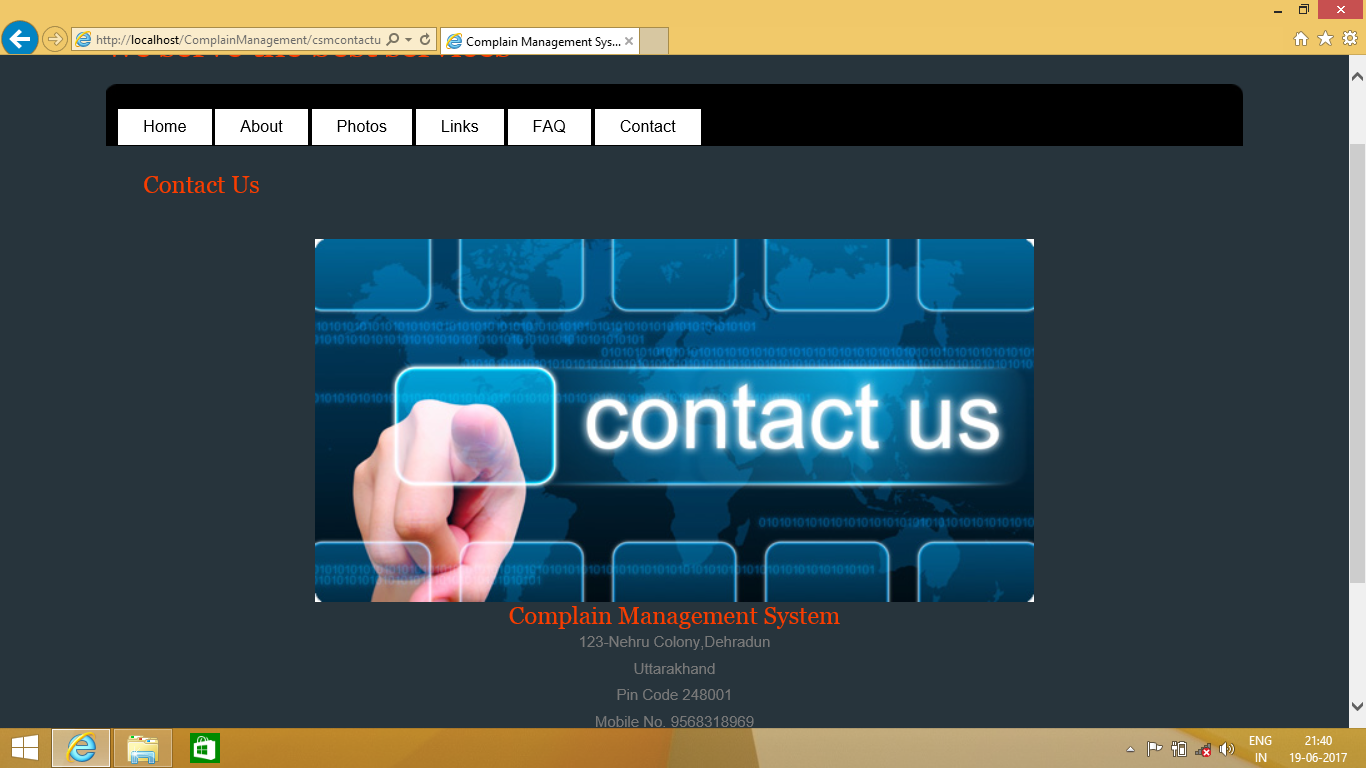
**Fig 3: Photos**

****

**Fig 4: Links**

****

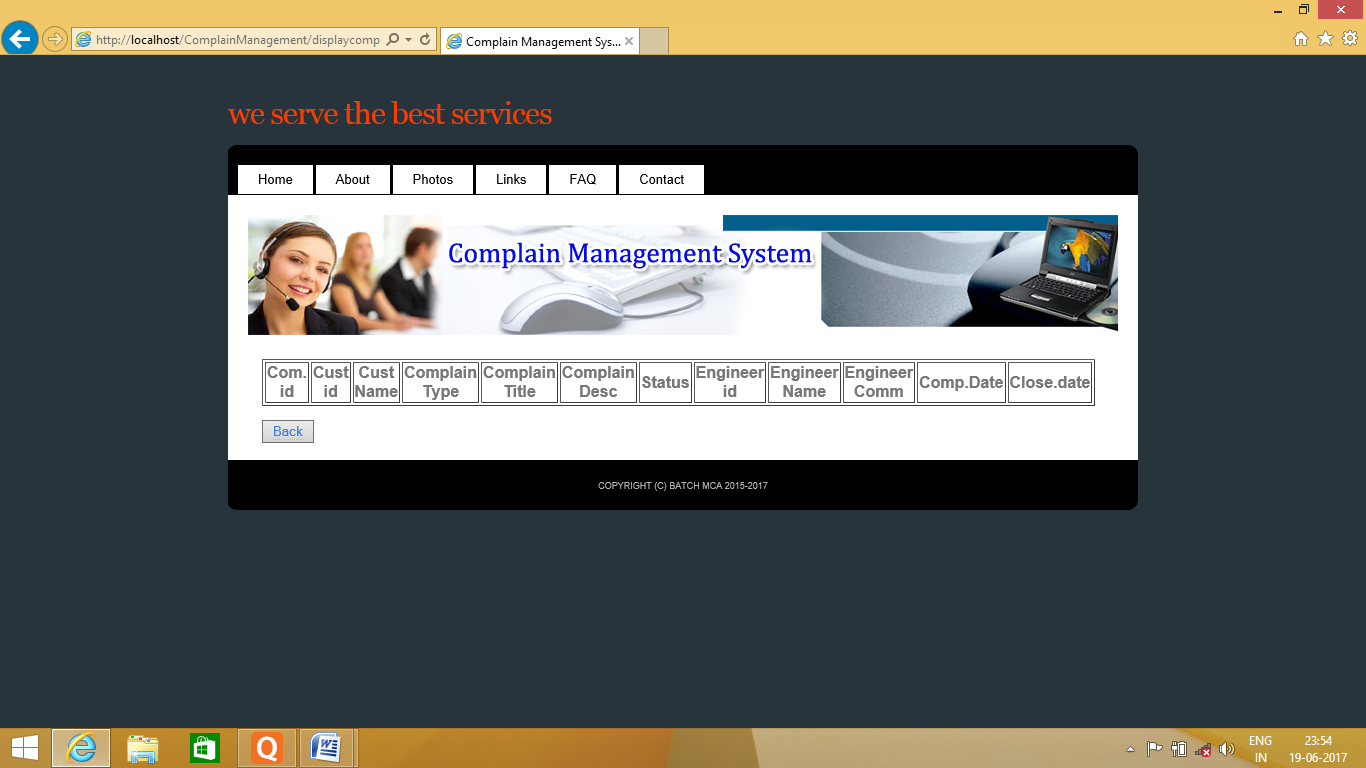
**Fig 5: FAQ**

****

**Fig 6: Contact Us**

****

**Fig 7: Admin Menu**

****

**Fig: 8 Complain**

****

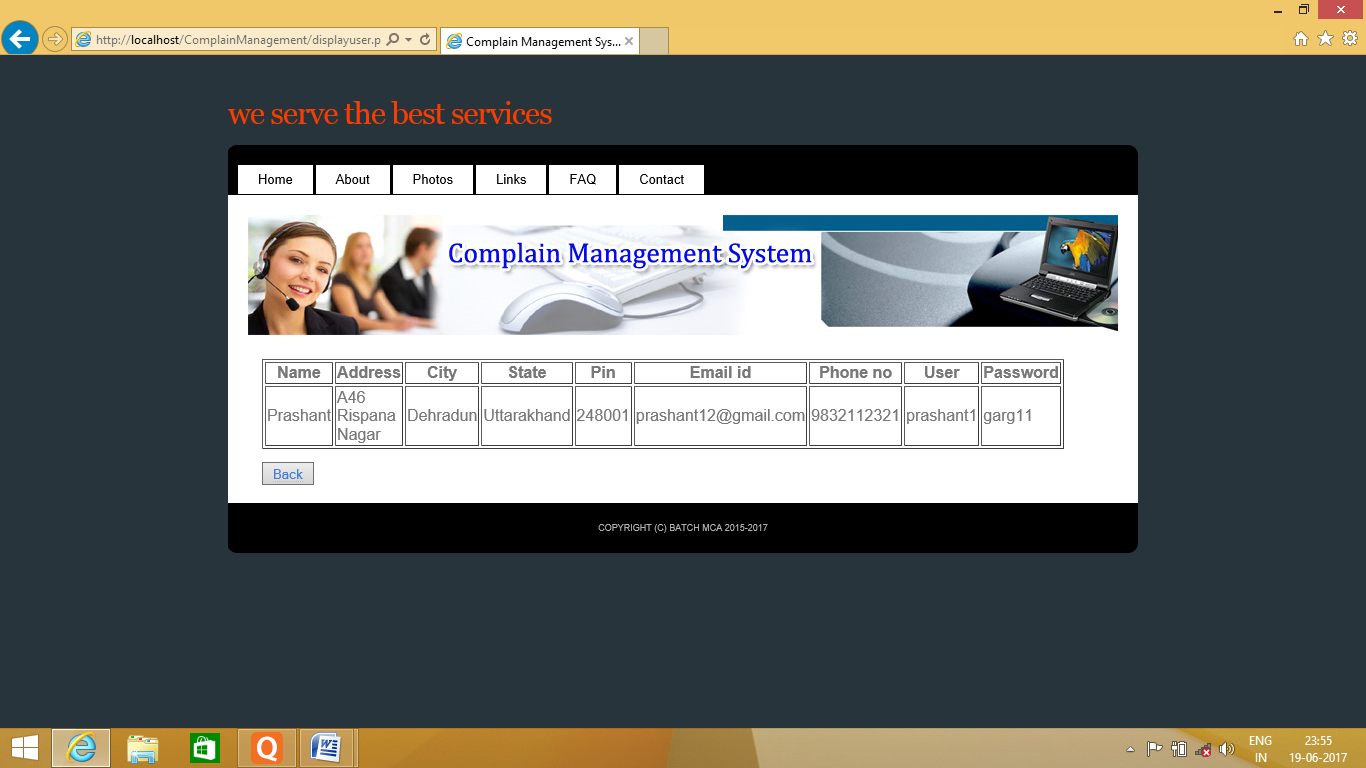
**Fig: 9 Engineer Page**

****

**Fig: 10 Engineer Update**

****

**Fig: 11 Engineer Id Update**

****

**Fig 12: User Update**

**TESTING & MAINTENANCE**

**TESTING:**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirement gaps, e.g., unrecognized requirements that result in errors of omission by the program designer. A common source of requirements gaps is non-functional requirements such as testability, scalability, maintainability, usability, performance, and security.

**The box approach**

Software testing methods are traditionally divided into white- and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

**White box testing**

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

The following types of white box testing exist:

• API testing (application programming interface) - testing of the application using public and private APIsCode coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

• Fault injection methods - improving the coverage of a test by introducing faults to test code paths

• Mutation testing methods

• Static testing - White box testing includes all static testing

**Test coverage**

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important functional points have been tested

Two common forms of code coverage are:

* Function coverage, which reports on functions executed
* Statement coverage, which reports on the number of lines executed to complete the test

**Black box testing**

Black box testing treats the software as a "black box"—without any knowledge of internal implementation.

**Specification-based testing**: Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks.

**Advantages and disadvantages**:

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and we shall receive," black box testers find bugs where programmers do not. On the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

**Grey box testing**

**Grey box testing** (American spelling: **gray box testing**) involves having knowledge of internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as grey box, because the input and output are clearly outside of the "black-box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test. However, modifying a data repository does qualify as grey box, as the user would not normally be able to change the data outside of the system under test. Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

**MAINTENANCE:**

The integral part of software is the maintenance part which requires accurate maintenance plan to be prepared during software development and should specify how users will request modifications or report problems and the estimation of resources such as cost should be included in the budget and a new decision should address to develop a new system and its quality objectives. The software maintenance which can last for 5–6 years after the development calls for an effective planning which addresses the scope of software maintenance, the tailoring of the post delivery process, the designation of who will provide maintenance, an estimate of the life-cycle costs.

**There are six software maintenance processes as:**

**1**. The implementation processes contains software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and the follow-up on product configuration management.

**2.** The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications.

**3.** The process considering the implementation of the modification itself.

**4**. The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.

**5**. The migration process (platform migration, for example) is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.

**6.** Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

There are a number of processes, activities and practices that are unique to maintainers, for example:

Transition: a controlled and coordinated sequence of activities during which a system is transferred progressively from the developer to the maintainer;

Service Level Agreements (SLAs) and specialized (domain-specific) maintenance contracts negotiated by maintainers;

Modification Request and Problem Report Help Desk: a problem-handling process used by maintainers to prioritize documents and route the requests they receive;

Modification Request acceptance/rejection: modification request work over a certain size/effort/complexity may be rejected by maintainers and rerouted to a develop.

**Scope and Limitations**

As systems grew more complex, it become evident that the goal of the entire system cannot be easily comprehended. Hence need for the requirements analysis phase arose. Now, for large software systems, requirements analysis is perhaps the most difficult activity and also the most error prone.

Some of the difficulty is due to the scope of this phase. The software project is imitated by the client needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify the requirements by tacking to these people and understanding there needs. In situations where the software is to automated a currently manuals process, most of the needs can be understood by observing the current practice.

**BIBLIOGRAPHY**

* PHP and MYSQLbWebDevlopment (Luke Welling &laura Thomson)
* PHP Solution (StpyanStefanov)
* Web Application Development (Jeffrey Winesett)
* CSS:The Missing Manual(David McFarland’s)
* JavaScript:The Good Parts(Douglas Crockford)
* Programming PHP(RasmusLerdorf)
* Solutions & Examples for PHP Programmers(David Sklar)
* HTML and CSS:Design and Build Website(Jon Duckett)

CONCLUSION

It is concluded that the application works well and satisfy the company and user. The application is tested very well and errors are properly debugged. The site is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested.

The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application, so that the web site functions very interactive and useful to existing application .The application satisfies both the company and user by eliminating more input. The speed of the transactions become more enough now.