**Fat-To-Fit**

**ABSTRACT**

We all know health is wealth. Being healthy is the first thing we need to keep in mind.

Because most of the time our attitude depends on how we feel. Being healthy and fit gives us

energy to do anything. Physical fitness is very necessary for a healthy and tension free life.

Physical fitness includes diet, exercise and sleep. These three basic things have their own

importance in each individual's life and everyone should be sensible with regard to these for a

healthy life. Our proposed “**Fat 2 Fit**” system is for those who run a gym business. Before

doing anything we did a decent research on major difficulties for gym owners. We examined

carefully how to make a huge registering system without failure as well as different functions

for different kinds of users depending on their privilege. The Gym Management requires a

system that will handle all the necessary and minute details easily and proper database

security accordingly to the user. They require software, which will store data about members,

employees, products, payroll, receipts of members & all transactions that occur in Gym.

**INTRODUCTION**

We all know health is a wealth. We do not need a fancy car, big apartment, a doctor degree

without a health. Being healthy is a first thing we need to keep in mind. Because most of time

our attitude depends on how we feel. Being healthy and fit gives us energy to do anything.

Physical fitness is very necessary for a healthy and tension free life. Physical fitness includes

diet, exercise and sleep. These three basic things have their own importance in each

individual's life and everyone should be sensible with regard to these for a healthy life.

The project entitle as Fat-to-Fit is developed to digitalize and create an automated

system. The system will perform the tasks like adding the new member to the gym, removing

the member or keeping the payment records and other stuff required in managing the gym

properly. Our proposed “Fat-to-Fit” is for those who run a gym business. Before doing

anything we did a decent research on major difficulties for gym owners. We examined

carefully about how to make a huge registering system without failure as well as different

functions for different kind of user depending on their privilege.

The online Fat-to-Fit system is user-friendly application. This automated system

makes all functionality easier for both owners and customers. It is very simple in design and

to implement. The system requirements are very low. System resources and the system will

work in almost all configurations.

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

An Existing system refers to the system that is being followed till now. The gym is working

manually. The current system is time consuming and also it is very costly, because it involves

a lot of paperwork. To manually handle the system was a very difficult task. But now-a-days

The following are the reasons why the current system should be computerized:

 ∙ To increase efficiency with reduced cost.

∙ To reduce the burden of paperwork.

∙ To save time management for recording details of each and every member and employee.

∙ To generate required reports easily.

**PROPOSED SYSTEM**

Here there are 4 users. They are The Admin, the Receptionist (gym instructor), the Physician

and the common User. Receptionist can add the details of a person who wish to join the gym

Their personal information including weight, height and phone number are collected. The

receptionist also provides timing for that person, when he can come to the gym.

Admin has more authority than the receptionist. He provides unique username and password

for the receptionist. He also has the right to delete or modify it. He even has the authority to

add the gym equipment to the software. He can also modify it. Finally when that person

wishes to leave the gym, his/ her present weight and height will be compared to his old height

and weight. He can even store the details of the medicine information which are in the gym

warehouse. He can even buy it from other medical shop and can be stored in the database so

that any information needed can be retrieved easily.

The main highlight of this application is that the common user can have a look at how their

body gets transformed after the course. This body transformation view is achieved through

uploading photos and via weight and BMI data.

**MODULE DESCRIPTION**

**MODULES**

1. ADMIN

Employee

* Register

Physician

* Register

User

* List
* Allot to gym instructor

Gym requirements

* Add
* Update

Complaints

* View
* Reply

Batches

* Add

Bill

* Generate

Attendance

* View

1. GYM INSTRUCTOR

Batch

* Add Timing

Attendance

* Add
* View

Diet chart

* Add
* Update
* Monitor

Transformation

* Update

First aid and healthy food

* Add

C. PHYSICIAN

Medicine

* Add
* Update

User

* View health details
* Approve Appointments
* Interacts

Medical doubts

* View
* Reply

D. USER

Diet chart

* View

Health details

* Add

Gym equipment’s

* View

Profile

* View
* Register

Healthy foods

* View
* Buy

Physician

* Take Appointments
* Interact

Transformation

* View

Medical doubts& Complaints

* Add

Attendance

* View

Bill details

* View

**FEASIBILITY STUDY**

A feasibility study is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project. Feasibility Study is performed to choose the system that meets the performance requirements at least cost. The most essential tasks performed by a Feasibility Study are the identification and description of candidate systems, the evaluation of the candidate systems and the selection of the best of the candidate systems. The best system means the system that meet performance requirements at the least cost. The most difficult part of a Feasibility Study is the identification of the candidate systems and the evaluation of their performances and costs. The new system has no additional expense to implement the system. The new system has advantages such as we can easily access files from any client in the Network, accurate output for accurate input and this application is more user friendly. We can use this application not only in this organization but also in other firms. So it is worth solving the problem.

**Technical Feasibility**

Technical Feasibility study is performed to check whether the proposed system is technically feasible or not. Technical feasibility centers on the existing computer system (hardware, software, etc) and to what extent it can support the proposed addition. This involves financial consideration to accommodate technical enhancement. This system is technically feasible. All the data are stored in files. The input can be done through dialog boxes which are both interactive and user friendly. Hard copies can be obtained for future use, by diverting the documents to a printer. Windows serves as the platform for the new system.

**Economical Feasibility**

Economical Feasibility Study is the most frequently used method for evaluating the effectiveness of a candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with cost. This analysis phase determines how much cost is needed to produce the proposed system. This system is economically feasible since it does not require any initial setup cost, as the organization has required machines and supporting programs for the application to execute itself. It does not need additional staffing requirements.

**Operational Feasibility**

Operational Feasibility study is performed to check whether the system is operationally feasible or not. Using command buttons throughout the application programs enhances operational feasibility. So maintenance and modification is found to be easier.

**Legal Feasibility**

Determines whether the proposed system conflicts with legal requirements or not. e.g.: A data processing system must comply with the local Data Protection Acts.

**Schedule feasibility**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is.

**Resource Feasibility**

This involves questions such as how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, etc. Contingency and mitigation plans should also be stated here.

**SYSTEM REQUIRMENTS**

**HARDWARE REQUIREMENTS:**

• System : Pentium IV 2.4 GHz.

• Hard Disk : 40 GB.

• Floppy Drive : 1.44 Mb.

• Monitor : 15 VGA Colour.

• Mouse : Logitech.

• Ram : 512 Mb.

**SOFTWARE REQUIREMENTS:**

• Operating system : Windows 7 and above

• Coding Language : PHP

• Data Base : SQLite

**SYSTEM DESIGN AND DEVELOPMENT**

The most creative and challenging phase of the life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations of the candidate system. The design may be defined as “the process of applying various techniques and principles for the purpose of defining a device, a process or a system with sufficient details to permit its physical realization”.

The designer’s goal is how the output is to be produced and in what format. Samples of the output and input are also presented. Second input data and database files have to be designed to meet the requirements of the proposed output. The processing phases are handled through the program Construction and Testing. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation. The importance of software design can be stated in a single word “Quality*”*. Design provides us with representations of software that can be assessed for quality. Design is the only way where we can accurately translate a customer’s requirements into a complete software product or system. Without design we risk building an unstable system that might fail if small changes are made. It may as well be difficult to test, or could be one who’s quality can’t be tested. So it is an essential phase in the development of a software product.

**Design process**

The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from a user-oriented document to document oriented to the programmers or database personnel. System design goes through to phase of development:

* Logical Design
* Physical Design

The dataflow diagram shows the logical flow of the system and defines the boundaries of the system. For a candidate system, it describe the inputs(source), output(destination), database(file) and procedures(dataflow), all in a format that meets the users requirement in logical design we specifies the user’s needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources.

Following logical design is physical design. This produces the working system by defining specification that tell programmers exactly what the candidate system must do, in turn we write the necessary programs or modifies the software package that accept input from the user, then perform the necessary operation through logical system design is one important phase of system design, the dataflow diagram is the logical flow of a system and defines the boundaries of the system, for a candidate system it describes the inputs or source, outputs or destination, database or data stores and procedures all in a format that meets the user needs. When analysts prepare the logical system design, they specify the user needs at level of detail that virtually. It determines the information flow into and out of the system and the required data resources. The logical system design covers: Reviews the current physical system its dataflow, file content, volumes, frequency etc.

Preparing output specification that determines the format, content and frequency of reports including terminal specification and location. Prepares input specification format, content and the most if the input functions. This includes determining the flow of the document from the input data source to the detailed output location. Prepares edit security and control specification , this includes specifying the rules for edit correction backup procedures and the controls that ensure processing file integrity specifies the implementation plan.

Prepare the logical design walks through the information flow output, input and controls and implementation plan reviews benefits, costs, target rates and system constraints the existing file and procedure reports.

**About the Tools**

In this project I am started with the web-based application. To do that initially we choose the programming language as Html, php based web application, front end is made with php and for the back end database layer MySQL. I have used responsive design method for developing User Interface, Which is adaptable for any Viewport.

**PHP**

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHPcode in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) or HTML5 [markup](https://en.wikipedia.org/wiki/Markup_language), or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), [web content management systems](https://en.wikipedia.org/wiki/Web_content_management_system) and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)) in the web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) [executable](https://en.wikipedia.org/wiki/Executable). The [web server](https://en.wikipedia.org/wiki/Web_server) software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated [web page](https://en.wikipedia.org/wiki/Web_page). PHP code may also be executed with a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) and can be used to implement [standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface). PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHPcode in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere.

**Hyper Text Mark-up Language (HTML)**

HTML or Hyper Text Markup Language is a simple scripting language that is interpreted within a web browser. HTML helps to make web pages. HTML is the basis behind (almost) all web pages. Some tags in an HTML document determine the way certain text, such as titles, will be formatted. Other tags cue the computer to respond to the user's actions on the keyboard or mouse. An important tag is a link, which may contain the Uniform Resource Locator (URL) of another document. The URL can be compared to an address where a particular document resides.

The document may be stored on the same computer as the parent document or on any computer connected to the World Wide Web. The user can navigate from document to document simply by clicking on these links. HTML also includes mark ups for forms, which let the user fill out information and electronically send, or e-mail, the data to the document author and initiate sophisticated searches of information on the Internet, or order goods and services.

**MYSQL**

The following list shows the most important properties of MySQL. This section is directed to the reader who already has some knowledge of relational databases. We will use some terminology from the relational database world without defining our terms exactly. On the other hand, the explanations should make it possible for database novices to understand to some extent what we are talking about.

**Relational Database System:** Like almost all other database systems on the market,   
 MySQL is a relational database system.

**Client/Server Architecture:** MySQL is a [client/server system](http://searchdatamanagement.techtarget.com/feature/Understanding-and-comparing-six-types-of-processing-systems). There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

**SQL compatibility:** MySQL supports as its database language -- as its name suggests – SQL (Structured Query Language). SQL is a standardized language for querying and updating data and for the administration of a database. There are several SQL dialects (about as many as there are database systems). MySQL adheres to the current SQL standard (at the moment SQL:2003), although with significant restrictions and a large number of extensions.

**SubSELECTs:** Since version 4.1, MySQL is capable of processing a query in the form SELECT \* FROM table1 WHERE x IN (SELECT y FROM table2) (There are also numerous syntax variants for subSELECTs.)

**Views:** Put simply, views relate to an SQL query that is viewed as a distinct database object and makes possible a particular view of the database. MySQL has supported views since version 5.0.

**Stored procedures:** Here dealing with SQL code that is stored in the database system. Stored procedures (SPs for short) are generally used to simplify certain steps, such as inserting or deleting a data record. For client programmers this has the advantage that they do not have to process the tables directly, but can rely on SPs. Like views, SPs help in the administration of large database projects. SPs can also increase efficiency. MySQL has supported SPs since version 5.0.

**Triggers:** Triggers are SQL commands that are automatically executed by the server in certain database operations (INSERT, UPDATE, and DELETE). MySQL has supported triggers in a limited form from version 5.0, and additional functionality is promised for version 5.1.

**Unicode:** MySQL has supported all conceivable character sets since version 4.1, including Latin-1, Latin-2, and Unicode (either in the variant UTF8 or UCS2).

**User interface:** There are a number of convenient user interfaces for administering a MySQL server.

**Full-text search:** Full-text search simplifies and accelerates the search for words that are located within a text field. If you employ MySQL for storing text (such as in an Internet discussion group), you can use full-text search to implement simply an efficient search function.

# Normalization of Database

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anamolies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

**Normalization is used for mainly two purposes,**

* Eliminating redundant (useless) data.
* Ensuring data dependencies make sense i.e data is logically stored.

## Normalization Rule

Normalization rules are divided into the following normal forms:

1. First Normal Form
2. Second Normal Form
3. Third Normal Form
4. BCNF
5. Fourth Normal Form

### First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

### Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

### Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

### Boyce and Code Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

* R must be in 3rd Normal Form
* and, for each functional dependency ( X → Y ), X should be a super Key.

### Fourth Normal Form (4NF)

A table is said to be in the Fourth Normal Form when,

1. It is in the Boyce-Codd Normal Form.
2. And, it doesn't have Multi-Valued Dependency.

**Dataflow Diagrams**

The database may be defined as an organized collection of related information. The organized information serves as a base from which further recognizing can be retrieved desired information or processing the data. The most important aspect of building an application system is the design of tables.

The data flow diagram is used for classifying system requirements to major transformation that will become programs in system design. This is starting point of the design phase that functionally decomposes the required specifications down to the lower level of details. It consists of a series of bubbles joined together by lines.

* Bubbles: Represent the data transformations.
* Lines: Represents the logic flow of data.

Data can trigger events and can be processed to useful information. System analysis recognizes the central goal of data in organizations. This dataflow analysis tells a great deal about organization objectives are accomplished.

Dataflow analysis studies the use of data in each activity. It documents this finding in DFD’s. Dataflow analysis give the activities of a system from the viewpoint of data where it originates how they are used or hanged or where they go, including the stops along the way from their destination. The components of dataflow strategy span both requirements determination and system’s design. The first part is called dataflow analysis.

As the name suggests, we didn’t use the dataflow analysis tools exclusively for the analysis stage but also in the designing phase with documentation.

**Notations used in Dataflow Diagrams**

The logic dataflow diagrams can be drawn using only four simple notations i.e., special symbols or icons and the annotation that associates them with a specific system. Since the choice of notation we follow, does not affect impede or catalyze the system process; we used three symbols from YOURDON notation and one from Gain and Sarson notation as specified below.

**Element References Symbols**

Data Flow Process

Process/Function

Data Store

Source or Sink

**Description**:

Process: describes how input data is converted to output Data

Data Store: Describes the repositories of data in a system

Data Flow: Describes the data flowing between process, Data stores and external entities.

Sources: An external entity causing the origin of data.

Sink: An external entity, which consumes the data.

**Constructing a DFD**

Several rules of thumb are used in drawing DFDs:-

* Process should be named and numbered for easy reference.
* The direction of flow is from source to destination, although they may flow back to a source. One way to indicate this is to draw a long flow line back to the source. An alternative way is to repeat the source symbol as a destination.
* When a process is exploded into lower-level details, they are numbered.
* The names of data stores, sources, and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

A level 0 DFD, also called a context level, represents the entire software elements as a single bible with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e. Level 1 DFD. Any process, which is complex in Level 1, will be further represented into sub functions in the next level .i.e. Level 2 DFD is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data process, sources or destination.

The DFD is designed to aid communication. DFD shows the minimum contents of data stores. In order to show what happens within a given process, then the detailed explosion of that process is shown. The DFD methodology is quite effective, especially when the required design is unclear and the user and the analyst need a notational language for communication.

**Context Diagram:**

The top-level diagram is often called a “*context diagram”*. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

**Types of Dataflow Diagrams**

DFDs are two types:

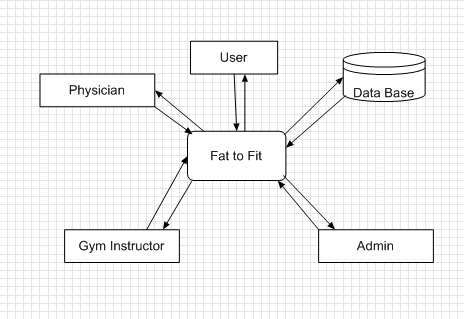
1. **Physical DFD**

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs shows actual devices, departments, people etc., involved in the current system

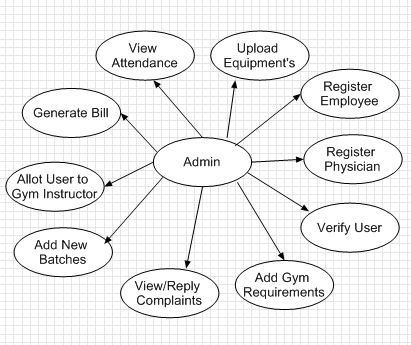
1. **Logical DFD**

mnLogical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts.

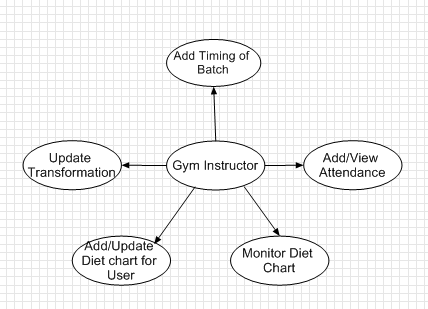
**Level-0**

****

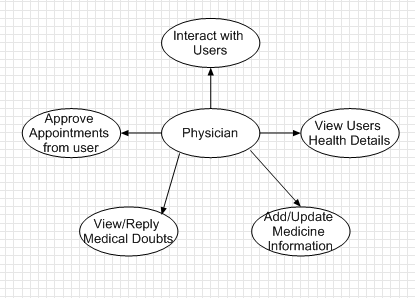
**Level 1 : Admin**

****

**Level 1 : Gym Instructor**

****

**Level 1 : Physician**

****

**Level 1 : User**

****

**INPUT DESIGN**

Input design is the process of converting user-oriented input into a computer based format. The goal of designing input is to make data entry as easy and free from error. Input to the system is entered through forms. A form is “Any surface on which information is to be entered, the nature of which is determined by what is already on that surface”. If the data going into the system is incorrect, then processing and output will magnify these errors. So design should ensure that form is accessible and understandable by the user.

End users are people who communicate to the system frequently through the user-interface, the design of the input screen should be according to their recommendations.

The following are the consideration given by the end-users for input design.

The screen should be user-friendly and easy to operate

Proper validation of inputs to be provide

The screen should be clear and enough information should be provided to guide the user to enter correct data

List of valid values for the field should be provide wherever possible\

The design decisions for handling input specify how data are accepted for computer processing. The design of inputs also includes specifying the means by which End-User and system operators direct the system in which the action to take. The goal of the input design is to make the data entry easier, logical and error free. Errors in the input data are controlled by input design. Complex name, figures etc. are avoided to make it user-friendly. Security is provided in necessary areas.

The application has been developed in a user friendly manner. The system accepts the needs from the user with and understandable dialogs. The screens have been designed in such a way that during the processing the cursor is placed in the position where the data must be entered. The input screens of the design are designed with standard layouts, colors and with appropriate controls like option button, check boxes etc. for making the data entry process easy, error free, fast with less strain. Data validation methods are used to check the data entered and to display appropriate error messages if any errors are found.

**OUTPUT DESIGN**

The most important thing about any system is what it produces. A System is judged to be a success or failure depending on whether its products are useful or not. So it is critical that we first specify what is required from the system. Once this had been done, we can concentrate on what is required from the system. Once this has been done we can concentrate on what is required to produce this output. In order to agree what results are to be produced by the system users are consulted to understand exactly what is required.

The main media available:

•Print used for reports and for a permanent listing of the file contents.

•Video display used for temporary output, usually responses to queries.

•Disk used for storing data files. These lies normally used for output and input.

Other factors to be considered when we are designing output are usage, quality and cost. These factors are closely related and we normally seek a compromise involving all three. For instance higher quantity generally cost more and a document to be used by the public needs to be better quality than once used within an organization. Thus were output is send can be divided into two broad groups internal and external. Internal usage refers to use by employees within organization, whereas external output is desired for people outside the organization.

The main requirement of an internal document is that it contains the necessary information for it to be useful. There is no need for fancy document or top quality printing or very high quality paper.

As long as the information is presented in a readable format the most important criterion as been satisfied. External documents, on the other hand, can play an important role in determining the public image of the organization. Thus the emphasis here is a presentation as well as usefulness. Later quality appearances etc. are given prime importance here.

Unfamiliar people use the external documents. So the terminology used must be simple. The higher the level of employee, the lesser the level of the employee, the lesser the details required in the report. How often a given report is needed or referenced can also influence is design. Some reports must be produced daily while others are less frequently required, in certain cases reports may be legal requirements. Sometime previous year report may be required so an appropriate output medium is selected for storing such reports.

**File Design**

A collection of related record makes up the file. The size of a file is limited by the size of memory or the storage medium. Two characteristics determine hoe file are organized, activity and volatility. File activity specifies the percentage of actual records, processed in a single run. File volatility address the properties of record change. 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 originated separately from other resources. Database is an integrated collection if data. The most significant difference between DBMS and other types of data management is the separation of data as seen by the programs and data as stored on the direct access storage devices. This is the difference between logical and physical data.

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

* Data integration
* Data integrity
* Data independence

The databases are implemented using a DBMS package. Each DBMS has unique characteristics and general techniques for database design.

**Code Design**

The goal of coding or programming phase is to translate the design of the system produces 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. For a given design, the aim is to implement the design in the best possible manner.

The coding phase affects both testing and maintenance profoundly. There are many different criteria for judging a program, including readability, size of the program, execution time, and required memory. The main objectives of the coding activity are: minimize the effort required to complete the program, minimize the number of statements, minimize the memory required, maximize the program clarity, and maximize the output clarity. Coding should be done in such a way that it is simple, easy to test, and easy to understand and modify.

Coding Style

Some special styles are used for the coding purpose, which results in producing simple, readable code and also helps for the future updating purpose of the same project.

Names

Variable names closely relate to the entity they represent and module names reflect their activity. This project is completely coded with PHP Codeigniter Naming Standards While developing the controllers, models and views, according to the Codeigniter standard all the views are named in the same name of the function that are used to accessing particular view, this is the security concern in this framework. Variables are named completely in lowercase and meaningful words.

Control Structures

In the case of control structures this project followed by Single-entry, single-exit constructs were used in the system.

Information Hiding

The ability to make changes in our implementation code without breaking the code of others who use your code is a key benefit of encapsulation. We want to hide implementation details behind a public programming interface. By interface, i mean the set of accessible methods your code makes available for other code to call—in other words, your code's API. By hiding implementation details, you can rework your method code (perhaps also altering the way variables are used by your class) without forcing a change in the code that calls your changed method

Module Interface

In this project, modules with complex interfaces are broken into multiple modules with simpler interfaces.

Program Layout

Proper indentation, blank spaces and parentheses are used to enhance the readability of the program.

Robustness

A program is robust if it does something planned even for exceptional conditions. A program should check for validity of inputs, where possible, and should check for possible overflow of the data structures .The proposed system handles such situations. The program won’t ‘crash’ or ‘core dumb’; it produce some meaningful message and exit gracefully.

**Database Design**

The general objective is to make information necessary, quick, inexpensive and flexible for the user. Database allows the data to be protected and organized separately from other resources. A database is a collection of interested data stored with minimum redundancy to serve many users quickly and efficiently. The general objectives considered in database design are controlled redundancy, case of learning and use, data independency, more information at low cost, accuracy and integrity, recover from failure, privacy and security performance.

In a database environment, the Database Management System (DBMS) is the software that provides the interface between the data file on a disk and the management, they differ in the way they structure data. The three types of data structures are hierarchical, network and relational. Here we use relational structuring in which all data and relationships are presented in a flat, two-dimensional table called a relation. A relation is equivalent to file.

Data structuring is refined through a process called normalization. Data are grouped into simplest way possible, so that later changes can be made with a minimum impact on data structures. Based on the requirements determined during the definition phase of project life cycle, the data elements describing the entity were determined. They are later submitted to normalization to remove redundancy and to optimize them.

Database objectives:

* Eliminate redundant data spontaneously
* Integrate existing data files
* Share data among all users
* Incorporate changes identifies easily and quickly
* Simplifies the use of data files
* Lower the cost of storing and retrieving data
* Improve accuracy and consistency
* Provide data security from unauthorized use
* Exercise central control over standards

Components of Database

Database is comprised of the following components, which are either data or the constructs to manage data:

* Field
* Tables
* Internal Database tables
* Database Constructs

Fields

Data is contained in information blocks called fields. Each separate piece of data is declared to contain a specific type of data as text, number, and date/time. Data types are used to protect the integrity of data. Various constraints may be placed on a field so that data integrity is ensured. Some of these constraints are unique, not null, and size. The assigning of constraints is taken care of and it forms the heart of the database design.

Tables

A table contains a group of fields of related information that define a single category. The table stores the data in fields. A set of fields that define one entry is called a record.

Internal Database Tables

In addition to the table defined some database also creates tables that keep track of what being done with the data.

Database Constructs

Database constructs are used to organize the database tables. Optimize the database for quick access, create relationship between the tables and specify who can access the different types of data.

**TABLE DESIGN**

**Login Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constrains** | **Description** |
| Login\_Id | Int(11) | Primary key | Login\_Id |
| User Name | Varchar(50) | - | User Name |
| Password | Varchar(50) | - | Password |

**Instructor Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constrains** | **Description** |
| Ins\_Id | Int(11) | Primary key | Ins\_Id |
| Instructor\_Name | Varchar(50) | - | Instructor\_Name |
| User Name | Varchar(50) | - | User Name |
| Password | Varchar(50) | - | Password |
| Address | Varchar(50) | - | Address |
| Mobile | int(50) | - | Mobile |
| Email | Varchar(50) | - | Email |

**Physician Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constrains** | **Description** |
| Phy\_Id | Int(11) | Primary key | Ins\_Id |
| Physician\_Name | Varchar(50) | - | Instructor\_Name |
| User Name | Varchar(50) | - | User Name |
| Password | Varchar(50) | - | Password |
| Address | Varchar(50) | - | Address |
| Mobile | int(50) | - | Mobile |
| Email | Varchar(50) | - | Email |

**Customers Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constrains** | **Description** |
| C\_Id | Int(11) | Primary key | Visited\_Id |
| Login\_Id | Int(11) | Foreign Key | Login\_Id |
| Customer\_Name | Varchar(50) | - | Customer\_Name |
| Fees | Int (50) | - | Fees |
| psw | Varchar(50) | - | psw |
| Mobile | int (50) | - | Mobile |
| Email | Varchar(50) | - | Email |

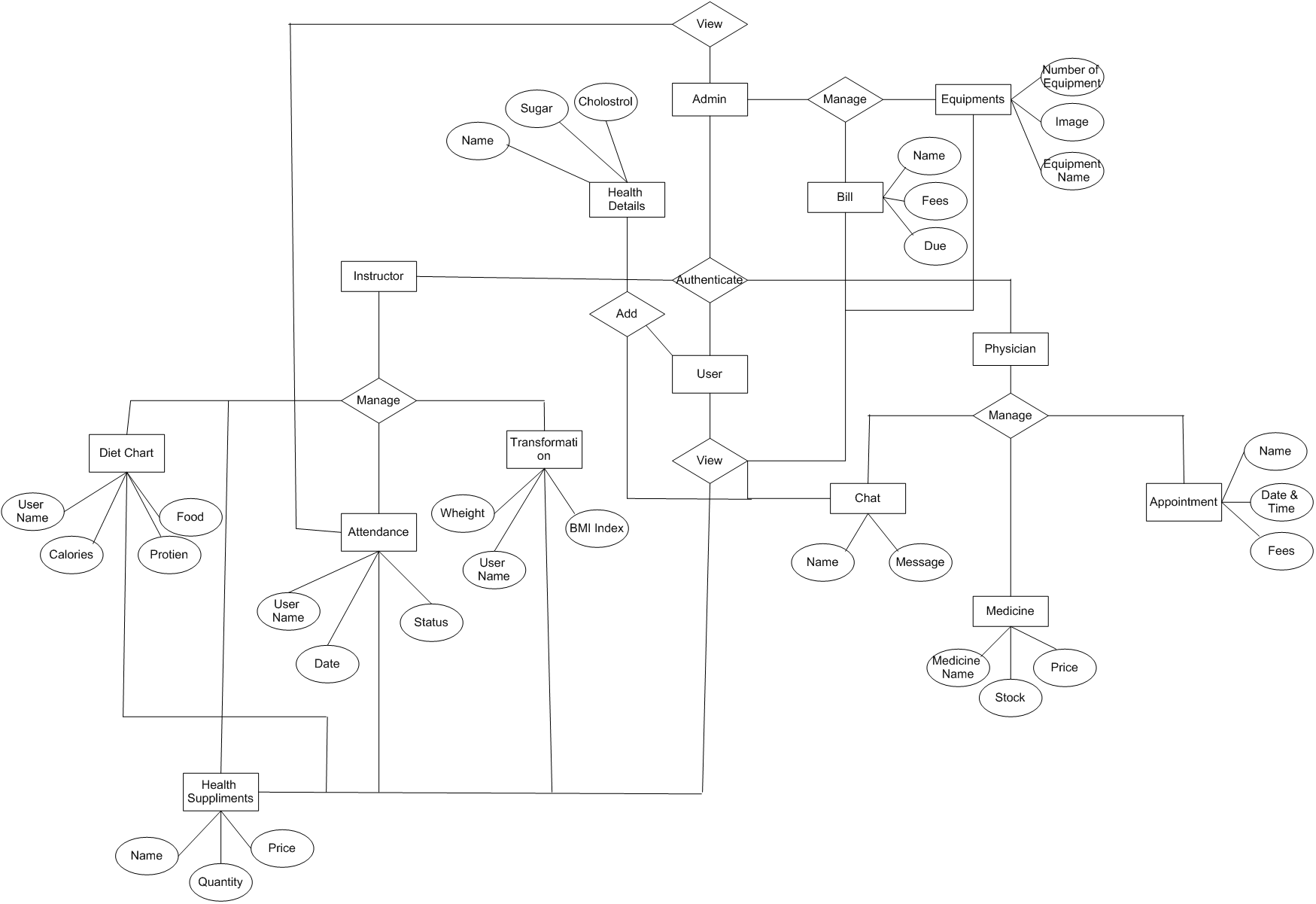
**Notification Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constrains** | **Description** |
| Notification\_Id | Int(11) | Primary key | Notification\_Id |
| Notification | Varchar(50) | - | Notification |

**Feedback Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constrains** | **Description** |
| Feedback\_Id | Int(11) | Primary key | Feedback\_Id |
| Name | Int(11) | - | Name |
| Feedback | Varchar(50) | - | Feedback |

**ER Diagram**

****

**SYSTEM TESTING AND IMPLEMENTATION**

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. For any software that is newly developed, primary importance is given to testing the system .It is the last opportunity for the developer over to the customers. Testing is the process by which a developer will generate a set of test data, which gives maximum probability of finding all types of errors that can occur in the software. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to a variety of tests: online response, volume, stress, recovery & security and usability tests. A series of testing are performed for the proposed system before the system is ready for user acceptance testing.

It is the process of exercising or evaluating a system by manual or automatic means to verify that it satisfies the specified requirements or to identify the difference between expected and actual results. The testing activities are aimed at convincing the customer through demonstration and actual use that the software is a solution to the original problem and that both the product and the process that created it are of high quality. It is also used to find and eliminate any residual errors from previous stages and the operational reliability of the system.

**Preparation of Test Data**

Software testing is a crucial element of software quality assurance and represents the ultimate review of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from abstract concepts to tangible implementation. The testing responsible for ensure that the product that has built performs the way that the detailed design documentation specifies.

**Goals and objectives**

The main purpose of testing an information system is to find the errors and correct them. The scope of system testing should include both manual and computerized operations. System testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls. System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted in accordance to the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner.

**Testing Objectives:**

* Testing is the process of executing a program, with the intent of finding so many errors as possible.
* A good test case is one that has a high probability of finding an as-yet-undiscovered error.
* A successful test is one that uncovers an as-yet-undiscovered error.

So, the main objective is to design tests that systematically uncover different classes of errors using minimum time and effort. Successful testing uncovers errors in software. It also shows that the software functions are working according to specifications. Also, the data collected during testing provides an indication of software reliability and software quality.

**Statement of scope**

The strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that result in the successful construction of software. The testing must co-operate with test planning, test case design, test execution and the resultant data collection and evaluation. A strategy for software testing must accommodate low level test and that are necessary to verify that a small code segment has correctly implemented as well as high level test that validate major system functions against user requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. A series of testing is performed for the proposed system before the system is ready for acceptance testing.

**Major constraints**

* All tests should be traceable to the customer requirements. According to the customer, the most severe defect is that which causes the program to fail to meet its requirements.
* Tests should be planned long before the actual testing begins. All tests should be planned and designed before any code is generated.
* The Pareto principle applies to software testing. The Pareto principle implies that 80% of all errors uncovered will likely be traceable to 20% of all program components. The problem is to isolate these suspect components and thoroughly test them.
* Testing should begin ‘in the small’ and progress towards testing ‘in the large’. The first tests focus on individual components. As testing progresses, focus shifts to integrated clusters of components and then finally to the entire system.
* Exhaustive testing is not possible. The number of path combinations in even a small program is very large. So it is not possible to test all these paths. But it is possible to test the program logic and ensure that all conditions have been met.
* To be most effective, testing should be conducted by an independent third party. The software engineering who created the program is not the best person to conduct tests for the software. So, in order to find the maximum number of errors in the software, an independent third party (who had no hand in developing the software) is preferred.

**There are several rules that can serve as testing objectives:**

* A good test is not redundant. Testing time and the resources are limited. So, a test that has the same purpose as another test need not be conducted. Every test should have a different purpose.
* A good test should be ‘best of breed’. There can exist a group of tests having the same intention. In such cases, only a subset of these tests is used. Thus, the test that has the highest chance of uncovering a whole class of errors should be used.
* A good test should neither be too simple nor be too complex. It is possible to combine a series of tests into one test. But this can lead to masking certain errors. Hence all the tests should be executed separately.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are subject to variety of tests on-line response, volume, stress, recovery and security and usability tests. Once the system performs flawlessly on artificial data, we switch to ‘Live Data’ or real data taken from the organization. A system is generally tested in a hierarchical fashion starting at the bottom and working up. First each program is tested; next a series of modules is tested; then each individual program with all its modules; finally the entire system consisting of a series of programs is tested. In this way, problems at the module level can be corrected before programs are tested and problems at the program level can be corrected before the entire system is used. A series of tests are performed before the system is ready for user acceptance testing.

**Testing methods**

The wide diffusion of Internet has produced a significant growth of the demand of Web-based applications with more and more strict requirements of reliability, usability, inter-operability and security. Due to market pressure and very short time-to-market, the testing of Web-based applications is often neglected by developers, as it is considered too time-consuming and lacking a significant payoff. This depreciable habit affects negatively the quality of the applications and, therefore triggers the need for adequate, efficient and cost effective testing approaches for verifying and validating them. Though the testing of Web-based applications (Web applications, in the remaining of the paper) shares the same objectives of ‘traditional’ application testing, in most cases, traditional testing theories and methods cannot be used just as they are, because of the peculiarities and complexities of Web applications. Indeed, they have to be adapted to the specific operational environment, as well as new approaches for testing them are needed.

In common a web based application is tested using:

* Performance testing
* Functionality testing
* Compatibility testing
* Accessibility testing
* Security testing
* Usability testing

**Performance testing**

Performance testing objective is to verify specified system performances (e.g. response time, service availability). It is executed by simulating hundreds, or more, simultaneous users accesses over a defined time interval. Information about accesses are recorded and then analyzed to estimate the load levels exhausting system resources. For Web applications, system performances is a critical issue because Web users don’t like to wait too long for a response to their requests, also they expect that services are always available. Performance testing of Web applications should be considered as an everlasting activity to be carried out by analyzing data from access log files, in order to tune the system adequately. Failures uncovered by performance testing are mainly due to running environment faults (such as scarce resources, or not well deployed resources, etc.), even if any software component of the application level may contribute to inefficiency.

**This testing Include:**

* Connection speed
* Load
* Stress

**Functional testing**

The functionality of the application like, calculation, business logic, validation links and navigation should be proper. In web based application the following functional tests are carried out.

* Links
  + Internal Links
  + External Links
  + Mail Links
  + Broken Links
* Forms
  + Field validation
  + Error message for wrong input
  + Optional and mandatory fields
* Database
  + Testing will be done on the database integrity.
* Cookies
  + Testing will be done on the client system side, on the temporary Internet files.

**Compatibility testing**

Compatibility testing will have to uncover failures due to the usage of different Web server platforms or client browsers, or different releases or configurations of them. The large variety of possible combinations of all the components involved in the execution of a Web application does not make it feasible to test all of them, so that usually only most common combinations are considered. As a consequence, just a subset of possible compatibility failures might be uncovered. Both the application and the running environment are responsible for compatibility failures.

**Accessibility testing**

It can be considered as a particular type of usability testing whose aim is to verify that access to the content of the application is allowed even in presence of reduced hardware/ software configurations on the client side of the application (such as browser configurations disabling graphical visualization, or scripting execution), or of users with physical disabilities (such as blind people). In the case of Web applications, accessibility rules such as the one provided by the Web Content Accessibility Guidelines have been established, so that accessibility testing will have to verify the compliance to such rules. The application is the main responsible for accessibility, even if some accessibility failures may be due to the configuration of the running environment (e.g., browsers where the execution of scripts is disabled).

**Security testing**

The objective of security testing is to verify the effectiveness of the overall Web system defenses against undesired access of unauthorized users, as well as their capability to preserve system resources from improper uses, and to grant the access to authorized users to authorized services and resources. System vulnerabilities affecting the security may be contained in the application code, or in any of the different hardware, software, middle-ware components of the systems. Both the running environment and the application can be responsible for security failures. In the case of Web applications, heterogeneous implementation and execution technologies, together with the very large number of possible users, and the possibility of accessing them from anywhere may make Web applications more vulnerable than traditional ones and security testing more difficult to be accomplished.

**Usability testing**

Usability testing aims at verifying to what extend an application is easy to use. Usability testing is mainly centered on testing the user interface: issues concerning the correct rendering of the contents (e.g. graphics, text editing format, etc.) as well as the clearness of messages, prompts and commands are to be considered and verified. Usability is a critical issue for a Web application: indeed, it may determine the success of the application. As a consequence, the front end of the application and the way users interact with it often are the aspects that are devoted greater care and attention along the application development process. When Web applications usability testing is carried on, issues about the completeness, correctness and conciseness of the navigation along application are to be considered and verified too. This type of testing should be a continuing activity carried out to improve the usability of a Web application; techniques of user profiling are usually used to reach this aim. The application is mainly responsible for usability failures.

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing is presents an interesting anomaly for the software. Testing is vital to the success of the system. Errors can be injected at any stage during development. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. During testing, the program to be tested is executed with set of test data and the output of the program for the test data is evaluated to determine if the program is performing as expected. A series of testing are performed for the proposed system before the system is ready for user acceptance testing. The testing steps are:

* Unit Testing
* Integration Testing
* Validation testing
* Output Testing
* Acceptance Testing

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of the software design, the module this is known as module testing. Since the proposed system has modules the testing is individually performed on each module. Using the details design description as a guide, important control paths are tested to uncover errors within the boundary of the module. This testing was carried out during programming stage itself. In this testing step each module is found to be working satisfactorily as regards to the expected output from the module.

**Integration Testing**

Data can be test across an interface; one module can have adverse effect on another, sub function when combined may not produced the desired function. Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated within the interface.

**Validation Testing**

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

* The function or performance characteristics confirm to specifications and are accepted
* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

**Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output generator or displayed by the system under consideration is tested by asking the users about the format required by them. Here the output is considered in two ways: One is on screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. As far as hardcopies are considered it goes in terms with the user requirement. Hence output testing does not result any correction in the system.

**Acceptance Testing**

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required.

Two types of testing are white-box testing and black-box testing.

White-Box testing

White-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. White-box testing of software is predicted on close examination of procedural detail.

Black-Box testing

The black-box testing focuses on the functional requirements of the software. It helps to find out errors in incorrect or missing functions, interface errors, errors in data structures, performance errors and initialization and termination errors. The black-box testing is applied during the later stages for the functional requirement evaluation.

Implementation

System implementation is the final phase i.e., putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively. The system is implemented only after thorough checking is done and if it is found working in according to the specifications.

It involves careful planning, investigation of the current system and constraints on implementation, design of methods to achieve. Two checking is done and if it is found working according to the specification, major task of preparing the implementation are educating, training the users.

The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. The most important in implementation stage is, gaining the users confidence that the new system will work and be effective. The system can be implemented only after through testing is done. This method also offers the greatest security since the existing system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

Implementation Methods

There are several methods for handling the implementation and consists for changing from the old to the new computerized system. The most secure method for conversion from the old system is to run the old and new system in parallel .In this approach; a person may operate in the manual processing system as well as start operating the new computerized system.

Another commonly used method is a direct cut over the existing manual system to the computerized system. The change may be within a week or a day. This strategy requires planning. A working version of the system can also be implemented in one part of the organization and the changes can be made as and when required, but this method is less preference due to the loss of entire system. After the system is Implementation, a review should be conducted to determine whether the system is meeting expecting where improvements are needed.

Implementation is the process of bringing a developed system into operational use and turning it over to the user. Implementation includes all those activities that take place to convert from old system to new. At this stage the theoretical design is turned into a working system. The crucial stage in achieving a successful new system and giving confidence on the system for the users that will work efficiently and effectively.

* The implementation stage involves the following tasks:
* Careful planning
* Investigation of the current system and its constraints
* Design of methods to achieve the changeover
* Training of staff in the overall procedures
* Evaluation of changeover

Implementation Plan

Implementation plan includes a description of all activities that must occur to implement the new system and to put into operation. It defines the personal responsible for the activities and prepares a time chart for Implementation the system. The Implementation plan should anticipate possible problems and must be able to deal with them. The usual problem may be missing documents, missed data formats between current and new files, errors in data translation, missing data etc.

Training plats are an important element of the implementation plan. Their purpose is to ensure that all the persons who are associated with the computer based business system possess the necessary knowledge and skills.

**Operator training**

Operator training is completed in conjunction with its installation and checkout. Operators must become familiar with operational requirements of the new systems. Well prepared manual provide a ready reference to specific duties and step by step operation instruction.

**User Training**

After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. Even well designed and technically elegant systems can succeed or fail, because of the way they are operated and used. For this purpose user manuals are prepared and handled over to the user to operate the developed system.

Thus the users are trained to operate the developed system. Both the hardware and software securities are made to run the developed systems successfully in future. In order to put new application system into use, preparation of user and system documentation, conducting user training with demo, test run for some period to ensure smooth switching over the system are to be prepared.

**Documentation**

The documentation involves collecting, organizing, and maintaining complete record of programs. The documentation deals with the system department with maximum clarity. Each and every process is explained in detail. The various table used by the system with field details are provided. The system uses various kinds of forms to produce well-structured screen formats. These forms are also documented .the output generated by the system constitutes another part. Documentation of the software provides the following:

**Comments**

Comments are very useful in documenting a program. It is used to explain logic of the program. It should be used to improve the quality and understanding of the program. It should not be redundant, incorrect or incomplete.

**System Manuals**

A good software system must contain standard system manuals. In this the statement is clearly defined, specifies description, detailed flowcharts, and specimen of all input forms and printed outputs,

OPERATION MANUALS

A good software package is supported with a good operation manual to ensure the smooth running of the program.

The operation manual must contain the following information:

* Setup and operational details of each program.
* Loading and unloading procedures.
* Starting, running, and terminating procedures.
* List of error conditions with explanations.

**SYSTEM SECURITY**

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The terms system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behaviour.

Internet is a part of everyday life, web applications are an essential component of every business activity. Customers and trading partners expect fast, accurate and secure applications with robust functionality. Companies want sites that are easy to maintain and update, yet cost effective. Auditors and security officers want to ensure that the web applications are controlled and that there is strong data integrity. All of these requirements need to be blended to ensure that each web application meets the company’s goals, satisfies the customers and trading partners, and is secure and reliable.

**The major security issues of web applications are:**

**Types of unauthorized access**

* Network security
* Firewalls
* Routers
* Intrusion detection and monitoring
* Virus detection and monitoring
* Encryption
* Operating system security
* Business continuance and disaster preparedness
* Hacker sites
* Control and security checklists

This is the process to determine that an Information System protects data and maintains functionality as intended. The six basic security concepts are:

**Confidentiality**

A security measure which protects against the disclosure of information to parties other than the intended users that is by no means the only way of ensuring.

**Integrity**

A measure intended to allow the receiver to determine that the information which it receives has not been altered in transit or by other than the originator of the information. Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.

**Authentication**

A measure designed to establish the validity of a transmission, message, or originator. Allows a receiver to have confidence that information it receives originated from a specific known source.

**Authorization**

This is the process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorization.

**Availability**

Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.

**Non-repudiation**

A measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp.

**Data security**

The focus behind data security is to ensure privacy while protecting personal or corporate data. Data is the raw form of information stored as columns and rows in our databases, network servers and personal computers. This may be a wide range of information from personal files and intellectual property to market analytics and details intended to top secret. Encryption has become a critical security feature for thriving networks and active home users alike. This security mechanism uses mathematical schemes and algorithms to scramble data into unreadable text. It can only by decode or decrypted by the party that possesses the associated key.

Data security wouldn't be complete without a solution to backup your critical information. Though it may appear secure while confined away in a machine, there is always a chance that your data can be compromised. You could suddenly be hit with a malware infection where a virus destroys all of your files.

Someone could enter your computer and thieve data by sliding through a security hole in the operating system. Perhaps it was an inside job that caused your business to lose those sensitive reports. If all else fails, a reliable backup solution will allow you to restore your data instead of starting completely from scratch.

**User security**

User security lets your application use security rules to determine what it displays. It has two elements:

Ensures that a valid user is logged-in, based on an ID and password provided by the user. ColdFusion (or, in some cases if you use web server authentication, the web server) maintains the user ID information while the user is logged-in.

**Authorization**

Ensures that the logged-in user is allowed to use a page or perform an operation. Authorization is typically based on one or more *roles* (sometimes called groups) to which the user belongs. For example, in an employee database, all users could be members of either the employee role or the contractor role. They could also be members of roles that identify their department, position in the corporate hierarchy, or job description. For example, someone could be a member of some or all of the following roles such as Employees, Human Resources, Benefits, and Managers. You can also use the user ID for authorization.

**Authenticating users**

You can use either, or both, of the following forms of authentication to secure your ColdFusion application:

* Web server authentication, where the web server authenticates the user and does not allow access to the website by users without valid login IDs.
* Application authentication, where the ColdFusion application authenticates the user and does not allow access to the application by users without valid login IDs.

**POST IMPLEMENTATION**

A Post-Implementation Review (PIR) is an assessment and review of the completed working solution. It will be performed after a period of live running; some time after the project is completed. There are three purposes for a Post-Implementation Review:

* To ascertain the degree of success from the project, in particular, the extent to which it met its objectives, delivered planned levels of benefit, and addressed the specific requirements as originally defined.
* To examine the efficiency of all elements of the working business solution to see if further improvements can be made to optimize the benefit delivered.
* To learn lessons from this project, lessons which can be used by the team members and by the organization to improve future project work and solutions.

**System Evaluation**

The system evaluation involves the hardware and software as a unit. The hardware selection is based on performance categories. The evaluation phase ranks vendor proposal and determines the one suited to the user’s needs. It looks in to items such as price, availability and technical support.

In the operation phase, the system performance must be monitored not only to determine whether or not they perform as planned, but also to determine if they should be modified to meet changes in the information needs of the business.

In the evaluation phase, the first step adopted was to look at the criteria listed earlier and rank them in the order of importance. Three sources of information are used in evaluating hardware and software. They are benchmark program, experience of other users and product reference manuals.

**Maintenance**

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment. Maintenance covers a wide range of activities, including correcting, coding and design errors, updating documentation and test data and upgrading user support. Maintenance means restoring something to its original condition.

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. The maintenance phase of the software cycle is the time in which a software product performs the useful work. If the new information is inconsistence with the design specification, then changes have to be made. The importance of maintenance is to continue to bring the new system to standards.

The system should be maintained and upgraded according to the technological advancements. It ensures the data integrity, data control and security. The system must be protected from fire and other natural calamities. The backup copies of data must be maintained daily so that we can prevent the loss of data due to various reasons.

**Types of changes that can be encountered during the maintenance phase:**

**Corrective maintenance**

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects.

**Adaptive maintenance**

Over time, the original environment (CPU, Operating System, Business Rules, External Product Characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

**Enhancement maintenance**

As software is used, the user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

**Preventive maintenance**

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

**Activities of a Maintenance Procedure**

Maintenance activities begin where conversion leaves off. Maintenance is handled by the same planning and control used in a formal system projects. The maintenance staff receives a request for service from an authorized user, followed by a definition of the required modifications.

The source program and written procedures for the system acquired from the programming library. Program changes are then tested and submitted to the user for approval. Once approved, the modified documentation is filled with the library and a project completion notice is sent to the user, signaling the termination of the project .Although software doesn’t ware out like a piece of hardware, it ages and evenly fails to perform because of cumulative maintenance .A major problem with the software maintenance is its labour-intensive nature and therefore the likelihood of errors.

**FUTURE SCOPE**

The future of Fat-to-Fit system is that it makes the customers more attractive and makes them easier to join & explore the gym facilities.Fitness business Growth and Future of the Fitness Business in India.The fitness industry has evolved in all these years and fitness lovers have been offered with lots of new opportunities and trends to look forward to which has helped in making it grow. Another factor that is boosting this industry is the increased level of awareness amongst common man with regards to physical fitness and wellness. The first and foremost responsibility is to guide trainees to get into shape. First aid qualifications are beneficial for most positions. It is having less paperwork. When a project is behind schedule, corrective steps must be taken. Modern retail is estimated at 28 per cent of the total market and is expected to grow by per cent. The techniques of project planning are not a substitute for good management, but merely a tool to be used by managers to achieve better results.

**CONCLUSION**

The “**FAT TO FIT**” is successfully designed and developed to fulfilling the necessary requirements, as identified in the requirements analysis phase, such as the system is very much user friendly, form level validation and field level validation are performing very efficiently. The old manual system was suffering from a series of drawbacks. The present project has been developed to meet the aspirations indicated in the modern age.

**BIBLIOGRAPHY**

* GOOGLE
* WIKIPEDIA
* Fundamentals of Software Engineering.-Ragib mall.
* Database system project.-H.F Korth.