

## **8. COST ESTIMATION OF THE PROJECT**

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (PM).

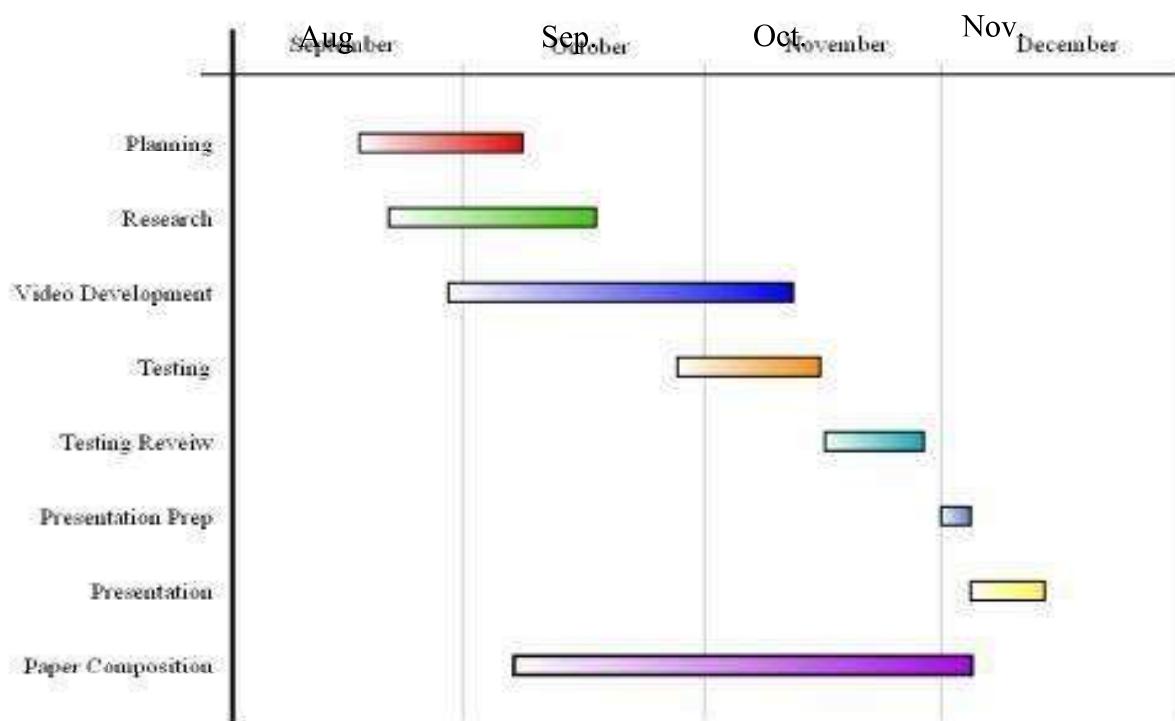
### **Total No. Of Persons Involved In This Project:**

1. Administrator

length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by colouring each milestone when completed. The project will start in the month of March and end after 4 months at the end of June.

## Gantt Chart



# **TABLE OF CONTENTS**

**QR CODE  
ACKNOWLEDGE  
SYNOPSIS**

## **CHAPTER – 1 INTRODUCTION OF PROJECT**

- 1.1 Introduction**
- 1.2 Objective of the Project**
- 1.3 Project Category**
- 1.4 Software and Hardware Requirements**
  - 1.4.1 JSP Introduction**

## **CHAPTER – 2 System Feasibility/Study**

- 2.1 System Development Life Cycle**
- 2.2 Feasibility Study**
- 2.3 Economic Feasibility**
- 2.4 Technical Feasibility**
- 2.5 Operational Feasibility**
- 2.6 Feasibility Report**

## **CHAPTER – 3 SYSTEM ANALYSIS**

- 3.1 Importance of computerised The Perfect Trainer [GYM MANAGEMENT SYSTEM]**
- 3.2 Principles of system analysis**

### **3.3 Methods used for gathering information**

## **CHAPTER – 4 SYSTEM DESIGN**

### **4.1 System design**

### **4.2 Physical design**

#### **4.2.1 Design methodology**

#### **4.2.2 Design overview**

#### **4.2.3 Process modelling**

### **4.3 ER Diagram**

### **4.4 Dataflow Diagram**

### **4.5 Modules of the project**

### **4.6 Feedback Module for GYM Management system**

## **CHAPTER – 5 CODING**

### **5.1 System development**

**Login source [WITH SCREENSHOT]**

**Add member source code [WITH SCREENSHOT]**

**Trainer details [WITH SCREENSHOT]**

**Delete member [WITH SCREENSHOT]**

**GYM MANAGEMENT SYSTEM [WITH SCREENSHOT]**

### **5.2 Code Efficiency**

### **5.3 Optimization of Code**

## **CHAPTER – 6 TESTING**

### **6.1 Testing**

**Defects and failures**

## **Compatibility**

## **Types of testing**

- **Static vs dynamic testing**
- **Unit testing**
- **Integration testing**
- **System testing**
- **Field testing**
- **Acceptance testing**
- **GANTT CHART**
- **PERT CHART**

**CHAPTER – 7 COST ESTIMATION OF THE PROJECT**

**CHAPTER – 8 SECURITY MEASURES**

**CHAPTER – 9 SCOPE OF FUTURE APPLICATION**

**CHAPTER – 10 CONCLUSION**

**CHAPTER – 11 BIBLIOGRAPHY**

Enrolment No: 2200054158  
BCSP-064

Name: SAURABH KUMAR  
The Perfect Trainer

---



**Project Proposal (Synopsis) of**  
**BACHELOR OF COMPUTER**  
**APPLICATION (BCA)**

On

**The Perfect Trainer**

To,

**Project Co-Ordinator (BCA)**

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

(RANGANATHAN BHAWAN, C BLOCK, NEAR CGHS DISPENSARY, BLOCK C,  
NARAINA VIHAR, NARAINA, NEW DELHI, DELHI 110028)

Submitted By:-

**NAME: SAURABH KUMAR**

**ENROLMENT NO: 2200054158**

Under Guidance Of :-

**HARSH JHA**



SCHOOL OF COMPUTER AND INFORMATION SCIENCES  
IGNOU, MAIDAN GARHI, NEW DELHI - 110 068

B30FR4167

**II. PROFORMA OF BCA PROJECT PROPOSAL (BCSP-064)**  
(Project's Title and Guide's Details)

Enrolment No.: 2200054158 ..... Regional Centre Code: 38039 .. Study Centre: RC-3 (38039)

**1. Name and Address of the student**

SAURABH KUMAR  
B-24 VIKAS KUNJ, VIKAS NAGAR, UTTAM NAGAR, NEW DELHI - 110059

E-mail: SINGHASHA9540@GMAIL.COM ..... Telephone No. 9540476142

**2. Title of the Project ..... THE PERFECT TRAINER**

**3. Name and Address of the Guide**

HARSH JHA  
84-C SAINIK VIHAR MOHAN GARDEN, UTTAM NAGAR, NEW DELHI - 110059

**4. Qualification of the Guide**

(Attach bio-data also)






Any other

(Note : i. All the above mentioned Degrees must have been awarded in Computer Science/IT only  
ii. A Guide should not guide more than 8 students of BCA at any point of time )

**5. Industrial / Teaching experience of the Guide (in Years) ..... 2 YEARS**

**6. Software Used for this Project: ..... HTML, JAVA, JAVA SCRIPT & MYSQL**

(Note :

1. Use of Visual Basic and MS-Access as Front End and Back End respectively is forbidden. But, you are permitted to use Visual Basic with other Software. Also, you can use MS-Access with other software
2. Use of C or C++ Programming Language for Project Related to Database Management is strictly forbidden.

Signature of the Student

Date:

28-June-2024

Signature of the Guide

Date:

29-06-2024

**Important:** 1. Attach this Proforma along with Guide's Bio-data and Project Synopsis in the Project Report.  
2. Not more than one student is permitted to work on a project.

For Office Use Only



Approved

Signature, Designation, Stamp of the  
Project Proposal Evaluator

Not approved

Date:

**Suggestions for reformulating the Project:** - Choose appropriate title  
- Properly draw DFD upto Level 2

- Synchronise ERD, DFD & DB tables
- Synchronise DFD & modules
- Follow BCSP 064 Project guidelines
- introduce feed back module

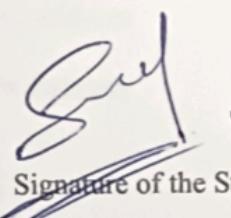
## **CERTIFICATE OF ORIGINALITY**

This is to certify that the project report entitled **The Perfect Trainer**

Submitted to **Indira Gandhi National Open University** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS ( BCA )**, is an original work carried out by **Mr. Saurabh Kumar**

Enrolment No.: **2200054158** under the guidance of **Mr. Harsh Jha**

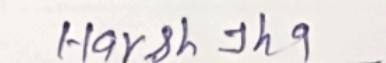
The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirement of any course of study.

  
Signature of the Student

Name and Address  
of the student :

**SAURABH KUMAR**

B-24 Vikas Kunj Vikas Nagar  
New Delhi - 110059

  
Signature of the Guide

Name, Designation and  
Address of the Guide

**HARSH JHA**

84-C Sainik Vihar Mohan Garden,  
Uttam Nagar New Delhi - 110059

Enrolment No.: **2200054158**

**BIO-DATA FORMAT FOR BCA PROJECT GUIDE**

1. Name (in BLOCK Letters) : ..... **HARSH JHA**

2. Date of Birth : ..... **28 SEPTEMBER, 1998**

3. Present Designation/  
Profession : ..... **Software Engineer**

4. Residential Address : ..... **84-C Sainik Vihar Mohan Garden, Uttam Nagar  
New Delhi - 110059**



**Phone Numbers** : (Resi) ..... (Mob) ..... +91 8826475102

E-mail address : [harshjha@kellton.com](mailto:harshjha@kellton.com)

5. Office Address : Lalwani House B, Sakore Nagar, Viman Nagar,  
Pune, Maharashtra - 411014

**Phone Numbers** : (Resi) ..... (Mob) .....

6. Academic Qualification (Attach the self attested photocopies of the educational qualification)

Degree	Branch/Specialisation	University	Year of Passing	Percentage of marks
<b>B.TECH</b>	ECE	Guru Gobind Singh IPU	2022	80.2%

#### **7. Teaching/Industry Experience:**

Teaching (UG/PG)/IT Job Undertaken)	Position/Designation	University/Institution	Topic Taught/Area of work executed	No. of years for teaching/working experience	Period From To
IT	Software Engineer	Kellton Technologies		2+	2022 to Present

Harsh Jhg

(Signature of the Guide)



# गुरु गोबिन्द सिंह इन्द्रप्रस्थ विश्वविद्यालय



## बैचलर ऑफ टैक्नोलॉजी (इलेक्ट्रॉनिक्स एण्ड कम्यूनिकेशन इंजीनियरिंग)

प्रमाणित किया जाता है कि हर्ष झा सुपुत्र/सुपुत्री विभूति झा विद्यार्थी महाराजा सूरजमल इंस्टीट्यूट ऑफ टैक्नोलॉजी को इस विश्वविद्यालय द्वारा जून 2022 में आयोजित तत्सम्बंधी परीक्षा प्रथम श्रेणी में उत्तीर्ण कर लेने के उपरान्त बैचलर ऑफ टैक्नोलॉजी (इलेक्ट्रॉनिक्स एण्ड कम्यूनिकेशन इंजीनियरिंग) की उपाधि प्रदान की गई।

## Guru Gobind Singh Indraprastha University

### Bachelor of Technology (Electronics and Communications Engineering)

This is to certify that **Harsh Jha** Son / Daughter of **Bibhuti Jha** a student of **Maharaja Surajmal Institute of Technology** is hereby awarded the degree of **Bachelor of Technology (Electronics and Communications Engineering)** on his/her having passed the Examination for the said degree in June, 2022 in the First Division.

विश्वविद्यालय के मुद्रांकन द्वारा प्रमाणित  
Given under the Seal of the University

(Prof. Gulshan Kumar)  
Controller of Examinations

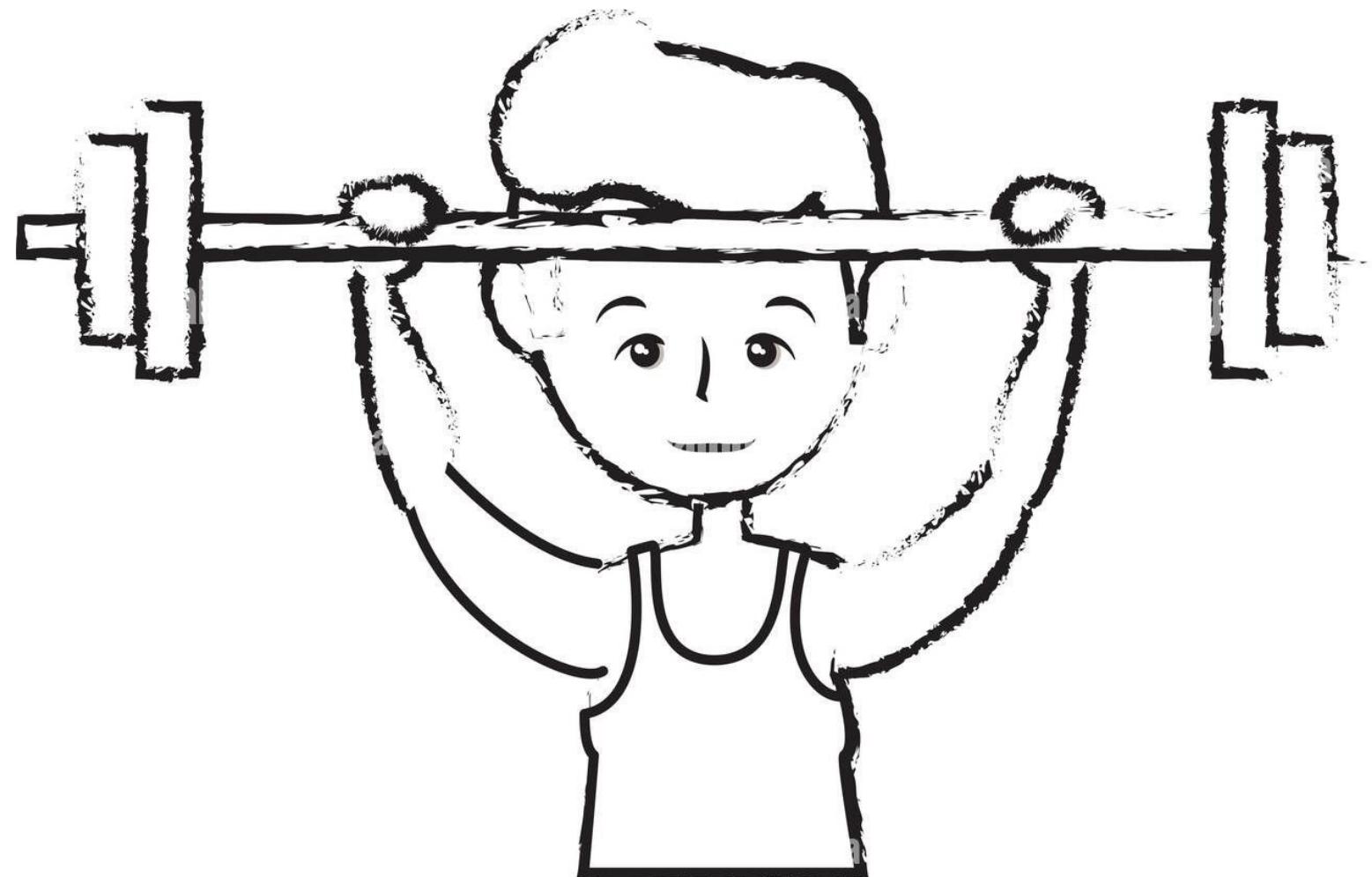


Date of Convocation : 06.06.2023  
दिल्ली (भारत)/Delhi (India)

(Prof. (Dr.) Mahesh Verma)  
Vice Chancellor



**TITLE OF THE PROJECT**



# **THE PERFECT TRAINER**

**[ GYM MANAGEMENT SYSTEM ]**

# **CONTENTS**

- 1. Introduction of project**
- 2. Scope and Objective of Project**
  - 1. Scope**
  - 2. Objective of Project**
- 3. Project Category**
- 4. Problem Definition of Project**
- 6. Module Description**
- 7. Planning and Scheduling Technique**
- 8. Tools/Platform, Hardware & Software Requirement**
- 9. Specification**
  - 1. Software**
  - 2. Hardware**
- 10. Data Flow Diagram**
- 11. Entiy Relationship Diagram**
- 12. Data Structure**
- 13. Testing of the Project**
- 14. Future scope and Further Enhancement**
- 15. Bibliography**

# INTRODUCTION

The Perfect Trainer is a GYM Management Software that is made for the Purpose of a quick and easy Communication between GYM Perosnal and the member. It is desied to seamlessly integrate all aspects of need of consumer then membership (Basic,intermidate,Advanced) Online bookings,payments with guaranteed security,Trial appointments. Locality Performance and also provide personal Trainer.

Our software provides lots of function such as details of various centers all over india which help to provide good quality of service to costumer.

This system allows for the integration of almost every possible area to reduce expenditure of valuable time searching perfect training centre and improve costumer experience in terms of providing benefits in a very simple way.

The members can find the area of interst based on their need to develop more strength and stamina in terms of their health and other activities.Based upon the several segments like most common weight loss,bodybuilding,elit performance and sport , a person can very easily go through the various option available to find their destination.

Fitness training balances five elements of good health .This flatorm may be only be starting to take the first step on road to fitness or you may get excited about exercise and want to improve your results Either way a,well rounded fitness training plan is essential .Basic five elements of fitness program includes aerobic fitness, strength training,flexibility and stretching to have a balanced routine.

# SCOPE & OBJECTIVES

## 1. SCOPE:-

The main scope of this project is to provide interactive platform for a GYM Lover. By having a clear picture of what you want and can achieve through gym management software, you can find the best solution for your needs that will benefit multiple areas of the business.

### 1. Member Management:-

A key objective of gym management systems is to simplify the entire member management process. The bloodline of any health club is a happy and sustained membership base. You can break the objective of member management into two categories: acquisition and retention. Gym management software should help you both market your business to new gym members and retain those you already have. Using gym software for member management will simplify all aspects of the process, such as check-ins, online registration, payments, and database management.

### 2. Gym Scheduling and Class Bookings:-

You should be able to easily schedule classes and manage all bookings and any other events you're hosting. A gym management system should link class bookings to your client-facing app so that your members can manage their classes and book time slots with their favorite instructor both online and offline. When investing in gym software or upgrading your existing system, this is a non-negotiable feature that will add to your value offering. Over time, it increases your member retention and attracts more customers who expect flexibility and complete convenience.

### 3. Manage Billing:-

Managing a gym tends to involve a ton of administrative tasks like contracts, invoices, billing, membership cancellations, and payments. Gym management software should consolidate all billing management into one location. When you can see an overview of your gym's finances, it can ease stress and save money in the long term. Finding out how to get the most out of your gym management software will mean that you can use it throughout the member journey. It will help cover everything from creating and signing contracts to automated billing and membership renewal reminders. A key objective of this type of software is to ease the workload on your team so they can focus their efforts on your members rather than being tied up in administrative tasks.

#### **4.Member Retention:-**

The right gym membership software solution will provide powerful tools to engage members and increase retention. By automating communication and interacting with members regularly, it increases the overall customer experience. Think about how you want to engage with your members; whether it's push notifications as a reminder to get into the studio or automated email promotions, you want to choose membership management that aligns with your goals.

#### **5.Email Marketing:-**

By using your gym management system for email marketing, you can keep clients engaged. Automation and personalization are key to increasing engagement while at the same time easing the number of resources you need to run email campaigns. Segment your audience and target your email list with content that is relevant and valuable. You can create monthly newsletters, nurturing campaigns, or send out quick updates whenever you need to inform your members of your latest news.

#### **6.Improve Member Experience:-**

An incredible member experience is an essential part of running a successful fitness business. You could have great equipment and an impressive facility, but it's your member experience that will set you apart from the competition. It's what makes your business unique and retains members over time. To create the best experience possible, it's important to streamline the customer journey and empower your members. That means booking, paying, and interacting with your brand should be as easy as possible. Members need to be able to book classes, pay for memberships and manage their gym schedule. Without this, the member experience will suffer. Nowadays, consumers expect a tech solution to streamline the ongoing relationship with your business.

#### **7. Reduce Costs:-**

Putting your time and resources into the right part of the business can reduce time-wasting and costs. Administrative tasks take up a lot of time for gym owners and managers. A gym management system should free up a lot of time so that your team can focus on what they do best; selling gym memberships, creating a great member experience, and producing fantastic member results. When you make decisions driven by data, you will likely invest your money and time much more effectively.

# **PROJECT CATEGORY**

## **Project Category:-**

This Project is OOPS/RDBMS based where below platform will be used during development and Designing Process.

<b>JAVA</b>	- As a font end interface
<b>MY SQL</b>	- As a back-End interface
<b>HTML, JAVA Script</b>	- As an Internet technology
<b>APACHE</b>	- As a Web server
<b>MS OFFICE PACKAGE</b>	- For Report

# PROBLEM DEFINITION OF PROJECT

In manual system we found that there are many problems exist, such as:

- 1) In the present system costumer has to approach various GYM academy to find a perfect training place and about their facilities , prices , timing. This often require a lot of time & efforts.
- 2) A costumer may spend most of the time to finding place of their interest and this can be tiring for them a lot of time.
- 3) Cost is also an important issue which trouble the customer So looking at a gather information within a short duration makes the task easy.
- 4) Facilites provided by different GYM operators can also different .Customer may not find them of thier interest quickly.

So reducing above mention prolem we have deceided to implement THE PERFECT TRAINER.Proposed system has following gaulities such as :

- 1) The proposed system is a web based application and maintains a centralised repository of all related information related to prices, facilities,location etc.
- 2) This allows one to easily access the relevant information and deciede which one to choose.
- 3) Constumer can select location according to their comfort and according to their need.
- 4) All points can be easily implemented in proposed project to make this system successful

# **MODULE DESCRIPTION**

## **1. Administrator module:**

This module provides administer related functionality. Administrator manages all information and has access right to add, delete, edit and view the data related to GYM Detail, Attendance, User Detail, Trainer Detail.

## **2. Staff Module:**

Staff and Admin both can mange the Member Deatil Provide the Gym at require Place and Require time.

## **3. Member Module:**

The in Member Module after the successfully registration Member access of option and mange the our profile. and select gym, time, area, trainer etc.

## **4. Package Module:**

The three type of package module available Basic, intermediate, Advanced.

## **5. Feedback module:**

feedback module is used to collect feedback from our customer. This module is mainly designed for Tourist. After enjoying tour, they can easily share their feedback and experience about tour.

# PLANNING AND SCHEDULING TECHNIQUE

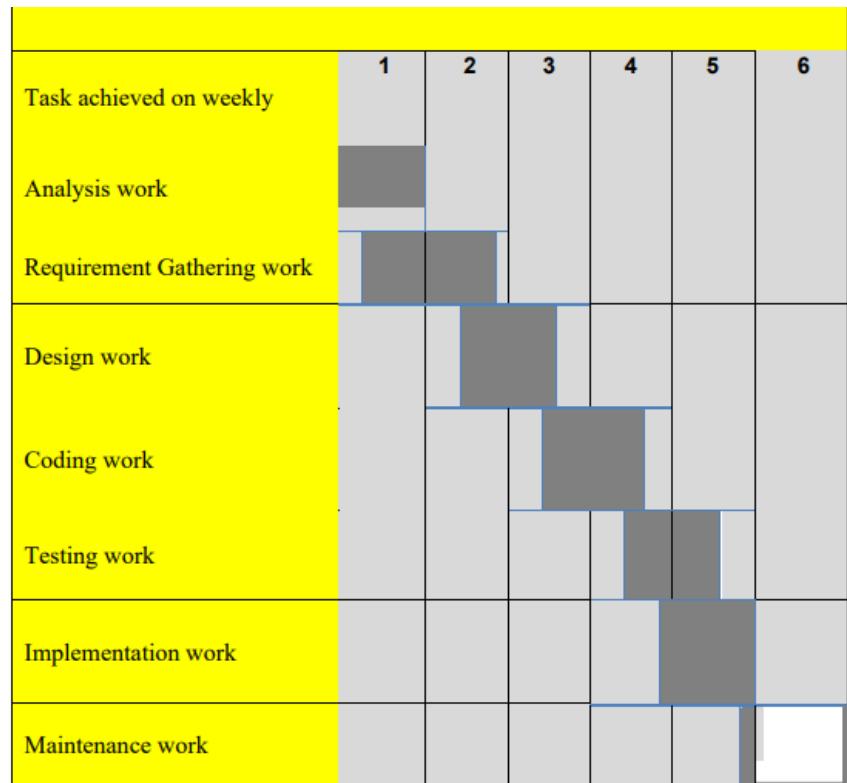


FIGURE: GANTT CHART

**TOOLS/PLATFORM, HARDWARE & SOFTWARE  
REQUIREMENT SPECIFICATION**

**SOFTWARE REQUIREMENTS:**

<b>Operating System</b>	<b>:</b> Window 10
<b>Technology</b>	<b>:</b> JAVA
<b>Database</b>	<b>:</b> My SQL

**HARDWARE REQUIREMENTS:**

<b>Hardware-Processor</b>	<b>:</b> Pentium - Pentium(minimum)
<b>RAM</b>	<b>:</b> 4GB
<b>Hard Disk</b>	<b>:</b> 500 GB
<b>Monitor</b>	<b>:</b> CRT/TFT - 17
<b>Key Board</b>	<b>:</b> PS2/USB
<b>Mouse</b>	<b>:</b> PS2/USB

# DATA FLOW DIAGRAM

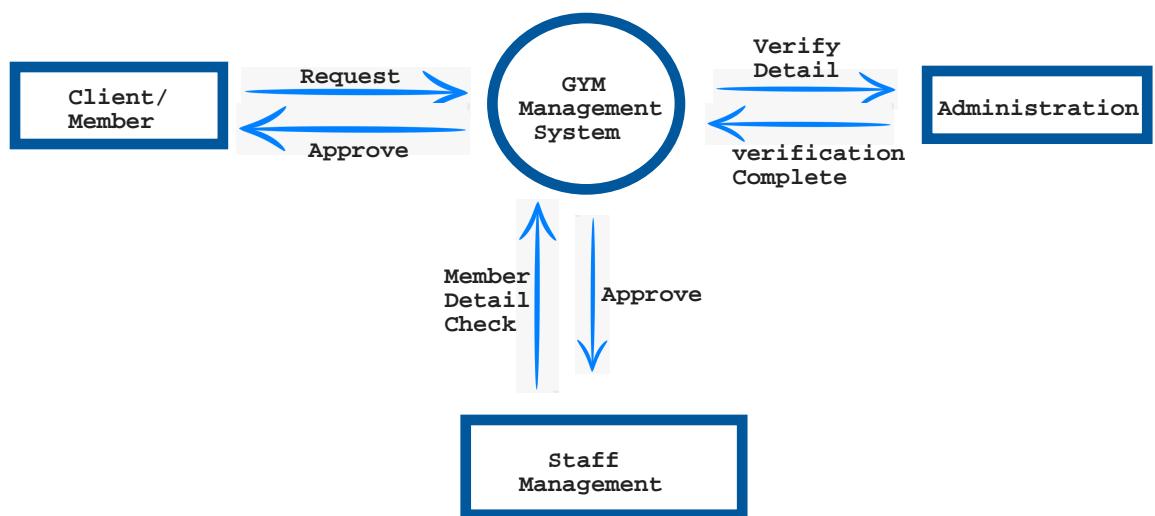


Figure:- DFD Level 0

# DATA FLOW DIAGRAM

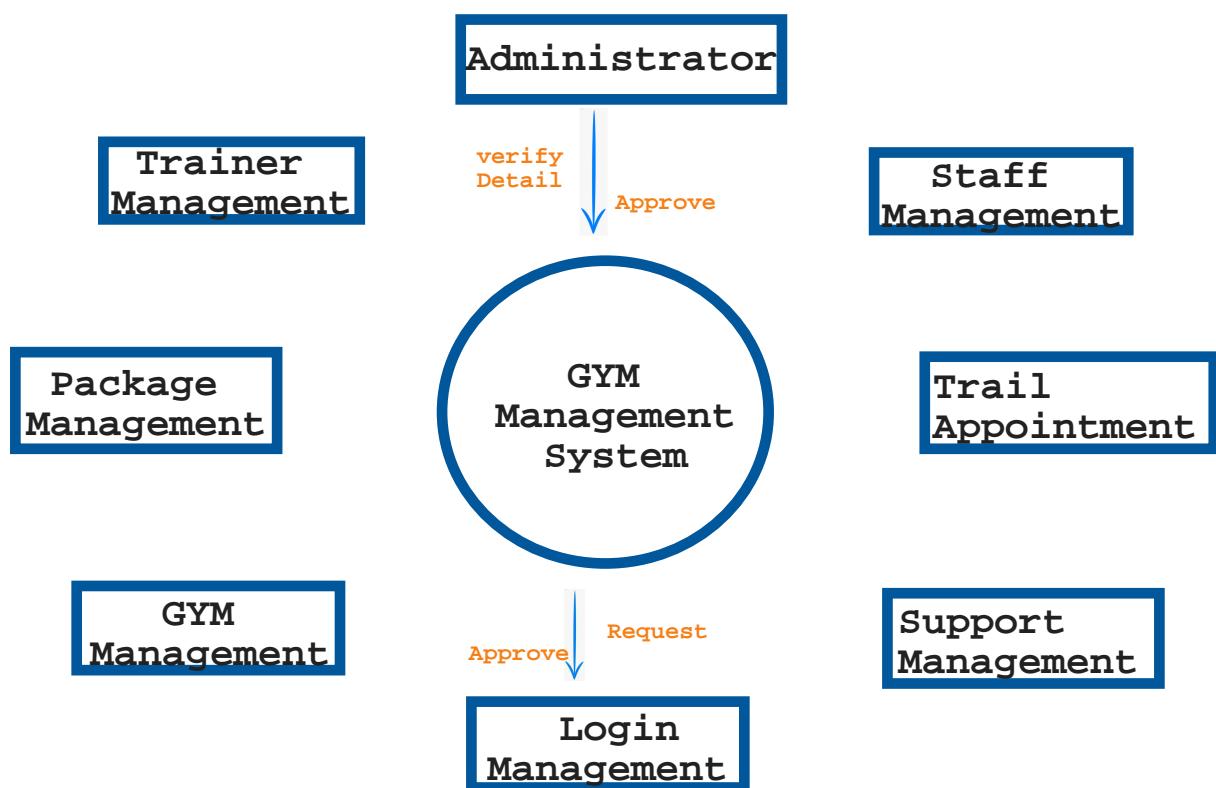


Figure:1 DFD LEVEL 1

# DATA FLOW DIAGRAM

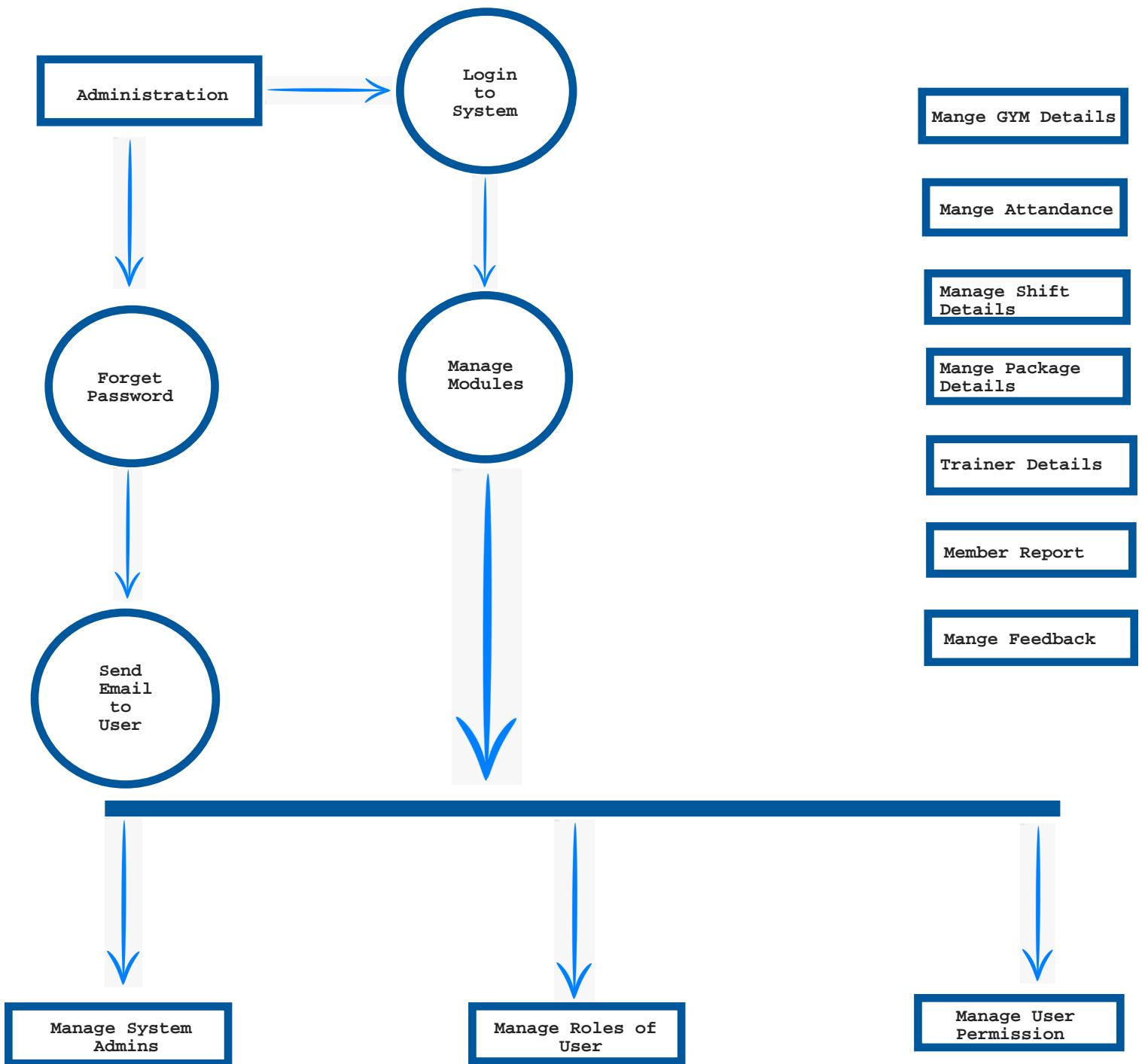
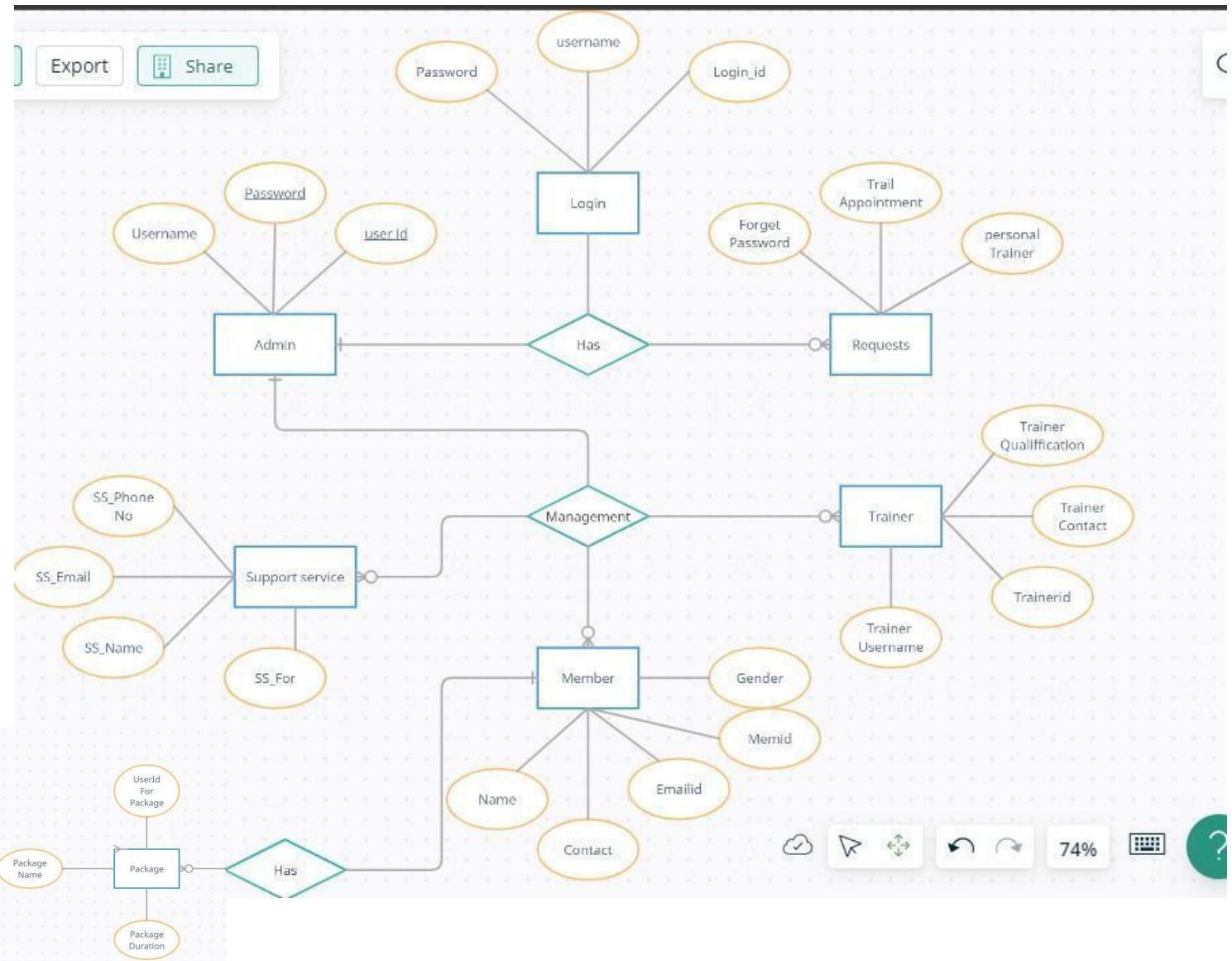


Figure: DFD Level 2

# ENTITY RELATIONSHIP DIAGRAM



# DATABASE STRUCTURE

**Administrator**

S. No.	Field Name	Data Type	Description
1.	A_name	Varchar (30)	Admin Name
2.	A_id	Varchar (10)	admin unique id
3.	A_pass	Varchar (10)	admin password
4.	A_email	Varchar (30)	admin Email Id

**Trainer Detail**

S. No.	Field Name	Data Type	Description
1.	to_id	Varchar (15)	Trainer Id
2.	to_location	Varchar (20)	Trainer Gym Location
3.	to_name	Varchar (30)	Triner Name
4.	to_address	Varchar (50)	Trianer Adress
5.	to_cont_no.	Varchar (15)	Trainer Contact no
6.	to_email	Varchar (20)	Trainer Email

**Member Deatil**

S. No.	Field Name	Data Type	Description
1.	Mem_Id	Varchar (15)	Member Id
2.	Mem_Name	Varchar (20)	Member Name
3.	Mem_Address	Varchar (30)	Member Address
4.	Mem_GYM Location	Varchar (50)	GYM Address
5.	Mem_Package	Varchar (15)	Member Package
6.	Mem_shift	Varchar (20)	Member Shift Time

**Trail Appointment**

S. No.	Field Name	Data Type	Description
1.	Name	Varchar (10)	Name of
2.	Contact No	Varchar (10)	Contact No of
3.	Email Id	Varchar (20)	Email Id of
4.	Feedback	Varchar (100)	Feedback of

**Feedback Module**

S. No.	Field Name	Data Type	Description
1.	Name	Varchar (10)	Name of Traveller
2.	Contact No	Varchar (10)	Contact No of Traveller
3.	Email Id	Varchar (20)	Email Id of Traveller
4.	Feedback	Varchar (100)	Feedback of Traveller

# TESTING OF THE PROJECT

For proper implementation and making successful project it should be properly tested by the testing methodologies. Actually, Software testing is the process of evaluating a software item to detect differences between given input and expected output. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

## Verification:-

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

## Validation:-

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

## Basic of Software testing:-

There are two basic software testing : black box testing and white box testing which is described below. But apart from that there are so many testing methods involved in software testing, some of them are:

- \* Unit Testing
- \* Integration Testing
- \* Functional Testing
- \* System Testing
- \* Stress Testing
- \* Performance Testing
- \* Usability Testing
- \* Acceptance Testing
- \* Regression Testing
- \* Beta Testing

## Black Box Testing:-

Black box testing is a technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

### **White Box Testing:-**

White box testing is testing technique that take into account the internal mechanism of a system. It is also called structural testing glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

There are some other test methodologies also present in soft-ware industries, so, other test method and respective result share at the time of development of proposed project.

# Limitations of the Project

## 1. Using a non-user-friendly dashboard:

One of the main reason your software can bring down your gym business is using non-user-friendly gym software. If the gym software you are using for your gym is difficult to use, non-understandable, and many of your customers can not fully use it, it can be unwholesome for your business.

## 2. Unnecessary repetitive updates:

If your gym software keeps updating, it can be another factor for downfall. Sometimes, you need the update so you can remove bugs from your gym software. But sometimes, your software keeps updating without any reason and changing its interference.

## 3. Not getting necessary features:

Suppose they want a feature to increase their workout efficiency, but your software is incompatible with such updates. Then it'll give your customers a reason to leave your gym and move forward.

## 4. Accuracy:

y:

Many software available that don't give accuracy to your business. They made non-accurate reports that can give you so much trouble. Because the owner knows how his business is going and how much profit he will make. But at the end of the month, if the software gives a false report, it will be like the pump damage.

# **Future Application of the Project**

THE PERFECT TRAINER is very interactive platform for manage the Gym Management system, in the gym system we have to integrate and coordinate between Trainer and Owner. And the coordination is successfully developed in this proposed project. It is very effective project to searching Perfect Trainer GYM.

Following are key points for future scope of the project:

- \* This Project should be properly introduced and advertised to the user, so that its popularity increases over time.
- \* This project should be fully Desktop based so that anyone can access using search engine.
- \* Coordination between Member and Administration should be enhanced to increase the usability of this project.
- \* In future we provide this system accessible as a mobile app format so that user can use it very easily.
- \* SMS and Email facility Provided to every tour operator, so that they get in touch with their client.
- \* Provide best services to the tourist and tour operator so that they happily use our system.

## BIBLIOGRAPHY

- o Introduction to Software Engineering (IGNOU)
- o Project Guide Booklet (IGNOU)
- o JAVA & My SQL Black Book (Dreamtech Press)
- o Software Engineering (Techmax/// Publication)

### ➤ WEB LINKS

1. [www.google.com](http://www.google.com)
2. [www.tutorialspoint.com](http://www.tutorialspoint.com)
3. [www.w3school.com](http://www.w3school.com)

# **PROJECT REPORT**

## **THE PERFECT TRAINER GYM MANAGEMENT SYSTEM**

# **1. INTRODUCTION**

## **1.1 INTRODUCTION**

GYM Personal and the member. It is designed to seamlessly integrate all aspects of need of consumer then membership (Basic, Intermediate, Advanced) Online bookings, payments with guaranteed security, Trial appointments. Locality Performance and also provide personal Trainer. Our software provides lots of function such as details of various centers all over India which help to provide good quality of service to customer. This system allows for the integration of almost every possible area to reduce expenditure of valuable time searching perfect training centre and improve customer experience in terms of providing benefits in a very simple way. The members can find the area of interest based on their need to develop more strength and stamina in terms of their health and other activities. Based upon the several segments like most common weight loss, bodybuilding, elite performance and sport, a person can very easily go through the various options available to find their destination. Fitness training balances five elements of good health. This platform may be only be starting to take the first step on road to fitness or you may get excited about exercise and want to improve your results. Either way a well-rounded fitness training plan is essential. Basic five elements of fitness program includes aerobic fitness, strength training, flexibility and stretching to have a balanced routine. Add an amount in your E-Wallet using your debit/credit card so that you don't have to carry your debit/credit card always with you and a special thing is that if any component has cash back offer then that cash back is directly added to your Wallet. The special thing about this website is that it shows you the best offer of every component, it also gives you the best offers associated with the component and seasonal offers also.

The best thing is that the users not only purchase any computer component from this website but also can avail great hardware and software related services provided by us. This website also provides users option to give their views about the usage of the purchased product which helps to provide better services in future. Either a user is novice or a tech guru, he/she is sure to enjoy shopping and get help from this website.

This website is developed using Java as its front-end and SQL Server as its backend.

## **1.2. OBJECTIVES OF THE PROJECT**

1.SCOPE: - The main scope of this project is to provide interactive platform for a GYM Lover By having a clear picture of what you want and can achieve through gym management software, you can find the best solution for your needs that will benefit multiple areas of the business.

1. Member Management: - A key objective of gym management systems is to simplify the entire member management process. The bloodline of any health club is a happy and sustained membership base. You can break the objective of member management into two categories: acquisition and retention. Gym management software should help you both market your business to new gym members and retain those you already have. Using gym software for member management will simplify all aspects of the process, such as check-ins, online registration, payments, and database management.

2.Gym Scheduling and Class Bookings: - You should be able to easily schedule classes and manage all bookings and any other events you're hosting. A gym management system should link class bookings to your client-facing app so that your members can manage their classes and book time slots with their favorite instructor both online and offline. When investing in gym software or upgrading your existing system, this is a non-negotiable feature that will add to your value offering. Over time, it increases your member retention and attracts more customers who expect flexibility and complete convenience.

3.Manage Billing: - Managing a gym tends to involve a ton of administrative tasks like contracts, invoices, billing, membership cancellations, and payments. Gym management software should consolidate all billing management into one location. When you can see an overview of your gym's finances, it can ease stress and save money in the long term. Finding out how to get the most out of your gym management software will mean that you can use it throughout the member journey. It will help cover everything from creating and signing contracts to automated billing and membership renewal reminders. A key objective of this type of software is to ease the workload on your team so they can focus their efforts on your members rather than being tied up in administrative tasks.

4.Member Retention: - The right gym membership software solution will provide powerful tools to engage members and increase retention. By automating communication and interacting with members regularly, it increases the overall customer experience. Think about how you want to engage with your members; whether it's push notifications as a reminder to get into the studio or automated email promotions, you want to choose membership management that aligns with your goals.

5.Email Marketing: - By using your gym management system for email marketing, you can keep clients engaged. Automation and personalization are key to increasing engagement while at the same time easing the number of resources you need to run email campaigns. Segment your audience and target your email list with content that is relevant and valuable. You can create monthly newsletters, nurturing campaigns, or send out quick updates whenever you need to inform your members of your latest news.

## **DRAWBACKS OF CURRENT MANUAL- SYSTEM**

- The current manual system has a lot of paper work and it difficult to maintain to paper of costumers.
- To maintain the records of costumers and service Trainers, is a Time-consuming job.
- With the increase in database, it will become a massive job to maintain the database.
- Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous purchases.
- The retrieval of records of the Items, purchasing details, Item categories will be a tedious job.
- Lack of security for the records, anyone disarrange the records of your system.

## **ADVANTAGE:**

I have designed the given proposed system in the JSP to automate **The Perfect Trainer**.

The following steps that give the detailed information of the need of proposed system are:

- Efficient management of members and their data
- Streamlined business operations and automation of routine tasks
- Improved communication and engagement with members
- Increased revenue and profitability through better business decisions and targeted marketing campaigns
- Enhanced member experience and retention through personalized communication and easy access to class schedules, billing, and membership details
- Improved staff scheduling and management, resulting in better employee morale and productivity
- Better inventory management and control, reducing waste and ensuring the availability of necessary supplies and equipment
- Enhanced security and safety by tracking member access and identifying potential risks

## **1.3 PROJECT CATEGORY**

Object Oriented Programming is a method of implementation in which programs are organized as cooperative collection of objects, each of which represents an instance of a class, and whose classes are all members of a hierarchy of classes united via inheritance relationship.

### **OOP Concepts**

Four principles of Object-Oriented Programming are

5. Abstraction
6. Encapsulation
7. Inheritance
8. Polymorphism

#### **Abstraction**

It is a technique which allows the hiding or elimination of useless information and focus essential. According to IEEE abstraction is defined as a view of a problem that extracts the essential information relevant to a particular purpose and ignore the remainder of the information.

#### **Encapsulation**

It is a mechanism that binds code and data together and keeps them secured from outside world. It is most striking feature of the class. The advantages of encapsulation are data hiding. Info hiding and implementation and independence. A change in the implementation to done without affecting the interface this leads to implementation to independence. Data encapsulation from external temping by user.

#### **Inheritance**

Inheritance is the process of reusability in java. In this one class can extend into another classes by including some additional method and attribute. The first existing class is called

super class and new class which extended the properly and method of existing classes and also called some additional methods and properties for itself is called derived class.

### **Polymorphism**

It is the ability to represent one thing in many forms, it means it improves the ability to use the same message to different object and then make them to behave differently. It is a

Greek word that consists of two words poly means “many” and morph means “form” so polymorph is equal to polymorphism

## **PROBLEM DEFINITION**

### **THE NEED OF NEW SYSTEM**

1. In the present system customer has to approach various GYM academy to find a perfect training place and about their facilities prices timing. This often requires a lot of time & efforts.
2. A customer may spend most of the time finding place of interest and this can be tiring for them a lot of time.
3. Cost is also an important issue which trouble the customer so looking at a gathered information within a short duration makes the task easy.
4. Facilities provided by different GYM operators can also differ. Customers may not find them of interest quickly.
5. The proposed system is a web based application and maintains a centralized repository of all related information related to prices facilities location etc.
6. Customer can easily select location according to their comfort and according to their need.

## **NEED**

I have designed the given proposed system in the JSP, MySQL2005 to automate the process. The following steps that give the detailed information of the need of proposed system are:

- **Performance:** During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the

Company system, the computerized system is to be undertaken. The computerized project is fully computerized and user friendly even that any of the members can see the report and status of the company.

- **Efficiency:** The basic need of this website is efficiency. The website should be efficient so that whenever a new user submits his/her details the website is updated automatically. This record will be useful for other users instantly.
- **Control:** The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.
- **Security:** Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

## **SOFTWARE & HARDWARE REQUIREMENTS**

### **HARDWARE**

<b>Processor</b>	:	i 3 Intel 2.4 GHz or above
<b>Memory</b>	:	4GB RAM or above
<b>Cache Memory</b>	:	128 KB or above
<b>Hard Disk</b>	:	3 GB or above [at least 3 GB free space required]
<b>Pen Drive</b>	:	5 GB
<b>Printer</b>	:	Laser Printer

### **SOFTWARE**

<b>Operating System</b>	:	Windows 8 10
<b>Font-End Tool</b>	:	JSP, Servlets, Java Script
<b>Back-End</b>	:	My Sql

### **FRONT END**

JSP programming tools are complete programming environments. It allows programmers to build a GUI program using the various on-screen controls such as buttons, text, menus, boxes etc. These controls are placed on a form and then the processing details related with each control are filled in.

In the business world, competitive strategies have become the order of the day to improve quality, cut costs and provide a high response customer service base. Most organizations today need to be market driven and do a lot of value addition to their products and services. This naturally calls for rational decision making, which requires information. Information Technology or IT provides that effective channel to support and implement this strategy. Client/Server is the technology that empowers the desktop, thus setting a trend for the way successful organizations will use technology in the next decade.

## **INTRODUCTION JSP**

### ***Adding dynamic content via expressions***

As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp. Of course, what makes JSP useful is the ability to embed Java. Put the following text in a file with .jsp extension (let us call it `hello.jsp`), place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello! The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Notice that each time you reload the page in the browser, it comes up with the current time.

The character sequences `<%=` and `%>` enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dynamic HTML pages that change in response to user actions or vary from user to user.

*Exercise:* Write a JSP to output the values returned by `System.getProperty` for various system properties such as `java.version`, `java.home`, `os.name`, `user.name`, `user.home`, `user.dir` etc.

## **Scriptlets**

We have already seen how to embed Java expressions in JSP pages by putting them between the `<%=` and `%>` character sequences.

But it is difficult to do much programming just by putting Java expressions inside HTML.

JSP also allows you to write blocks of Java code inside the JSP. You do this by placing your Java code between `<%` and `%>` characters (just like expressions, but without the `=` sign at the start of the sequence.)

This block of code is known as a "scriptlet". By itself, a scriptlet doesn't contribute any HTML (though it can, as we will see down below.) A scriptlet contains Java code that is executed every time the JSP is invoked.

Here is a modified version of our JSP from previous section, adding in a scriptlet.

```
<HTML>
<BODY>
<%
    // This is a scriptlet. Notice that the "date"
    // variable we declare here is available in the
    // embedded expression later on.
    system.out.println("Evaluating date now");
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

If you run the above example, you will notice the output from the `"System.out.println"` on the server log. This is a convenient way to do simple debugging (some servers also have techniques of debugging the JSP in the IDE. See your server's documentation to see if it offers such a technique.)

By itself a scriptlet does not generate HTML. If a scriptlet wants to generate HTML, it can use a variable called `"out"`. This variable does not need to be declared. It is already predefined for scriptlets, along with some other variables. The following example shows how the scriptlet can generate HTML output.

```
<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date" System.out.println(
    "Evaluating date now" );  java.util.Date
    date = new java.util.Date();
%>
Hello! The time is now
```

```

<%
// This scriptlet generates HTML output
out.println( String.valueOf( date ) );
%>
</BODY>
</HTML>

```

Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type [javax.servlet.jsp.JspWriter](#).

Another very useful pre-defined variable is "request". It is of type [javax.servlet.http.HttpServletRequest](#)

A "request" in server-side processing refers to the transaction between a browser and the server. When someone clicks or enters a URL, the browser sends a "request" to the server for that URL, and shows the data returned. As a part of this "request", various data is available, including the file the browser wants from the server, and if the request is coming from pressing a SUBMIT button, the information the user has entered in the form fields. The JSP "request" variable is used to obtain information from the request as sent by the browser. For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.) Let us modify the code as shown:

```

<HTML>
<BODY>
<%
// This scriptlet declares and initializes
"date" System.out.println( "Evaluating date now" );
java.util.Date date = new java.util.Date();
%>
Hello! The time is now
<%
out.println( date );
out.println( "<BR>Your machine's address is " );
out.println( request.getRemoteHost() );
%>
</BODY>
</HTML>

```

A similar variable is "response". This can be used to affect the response being sent to the browser. For instance, you can call `response.sendRedirect( anotherUrl )`; to send a response to the browser that it should load a different URL. This response will actually go all the way to the browser. The browser will then send a different request, to "anotherUrl". This is a little different from some other JSP mechanisms we will come across, for including another page or forwarding the browser to another page.

*Exercise:* Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.

## **Mixing Scriptlets and HTML**

We have already seen how to use the "`out`" variable to generate HTML output from within a scriptlet. For more complicated HTML, using the `out` variable all the time loses some of the advantages of JSP programming. It is simpler to mix scriptlets and HTML.

Suppose you have to generate a table in HTML. This is a common operation, and you may want to generate a table from a SQL table, or from the lines of a file. But to keep our example simple, we will generate a table containing the numbers from 1 to N. Not very useful, but it will show you the technique. Here is the JSP fragment to do it:

```
<TABLE BORDER=2>
<%
for ( int i = 0; i < n; i++ ) {
    %
    <TR>

    <TD>Number</TD>
        <TD><%= i+1 %></TD>
    </TR>
    %
}
%
</TABLE>
```

You would have to supply an `int` variable "n" before it will work, and then it will output a simple table with "n" rows.

The important things to notice are how the `%>` and `<%` characters appear in the middle of the `"for"` loop, to let you drop back into HTML and then to come back to the scriptlet.

The concepts are simple here -- as you can see, you can drop out of the scriptlets, write normal HTML, and get back into the scriptlet. Any control expressions such as a "while" or a "for" loop or an "if" expression will control the HTML also. If the HTML is inside a loop, it will be emitted once for each iteration of the loop.

Another example of mixing scriptlets and HTML is shown below -- here it is assumed that there is a Boolean variable named "`hello`" available. If you set it to true, you will see one output, if you set it to false, you will see another output.

```
<%
if ( hello ) {
    %
    <P>Hello, world
```

```

<%
} else {
%
<P>Goodbye, world
<%
}
%>

```

It is a little difficult to keep track of all open braces and scriptlet start and ends, but with a little practice and some good formatting discipline, you will acquire competence in doing it.

*Exercise:* Make the above examples work. Write a JSP to output all the values returned by `System.getProperties` with "<BR>" embedded after each property name and value. Do not output the "<BR>" using the "out" variable.

### **JSP Directives**

We have been fully qualifying the `java.util.Date` in the examples in the previous sections. Perhaps you wondered why we don't just import `java.util.*`;

It is possible to use "import" statements in JSPs, but the syntax is a little different from normal Java. Try the following example:

```

<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%
    System.out.println( "Evaluating date now" );
    Date date = new Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>

```

The first line in the above example is called a "directive". A JSP "directive" starts with `<%@` characters.

This one is a "page directive". The page directive can contain the list of all imported packages. To import more than one item, separate the package names by commas, e.g.

```
<%@ page import="java.util.*,java.text.*" %>
```

There are a number of JSP directives, besides the page directive. Besides the page directives, the other most useful directives are include and taglib. We will be covering taglib separately.

The include directive is used to physically include the contents of another file. The included file can be HTML or JSP or anything else -- the result is as if the original JSP file actually contained the included text. To see this directive in action, create a new JSP

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<%@ include file="hello.jsp" %>
</BODY>
</HTML>
```

View this JSP in your browser, and you will see your original `hello.jsp` get included in the new JSP. *Exercise:* Modify all your earlier exercises to import the `java.util` packages.

### **JSP Declarations**

The JSP you write turns into a class definition. All the scriptlets you write are placed inside a single method of this class.

You can also add variable and method declarations to this class. You can then use these variables and methods from your scriptlets and expressions.

To add a declaration, you must use the `<%!` and `%>` sequences to enclose your declarations, as shown below.

```
<%@ page import="java.util.*" %>
<HTML> <BODY>
<%!
Date theDate = new Date();
Date getDate()
{
    System.out.println( "In getDate() method" );
    return
theDate;
}
%>
Hello! The time is now <%= getDate() %>
</BODY>
</HTML>
```

The example has been created a little contrived, to show variable and method declarations.

Here we are declaring a Date variable `theDate`, and the method `getDate`. Both of these are available now in our scriptlets and expressions.

But this example no longer works! The date will be the same, no matter how often you reload the page. This is because these are declarations, and will only be evaluated once when the page is loaded! (Just as if you were creating a class and had variable initialization declared in it.)

*Exercise:* Modify the above example to add another function `computeDate` which reinitializes `theDate`. Add a scriptlet that calls `computeDate` each time.

**Note:** Now that you know how to do this -- it is in general not a good idea to use variables as shown here. The JSP usually will run as multiple *threads* of one single instance. Different threads would interfere with variable access, because it will be the same variable for all of them. If you do have to use variables in JSP, you should use *synchronized* access, but that hurts the performance. In general, any data you need should go either in the *session* object or the *request* object (these are introduced a little later) if passing data between different JSP pages. Variables you declare inside *scriptlets* are fine, e.g. `<% int i = 45; %>` because these are declared inside the local scope and are not shared.

## JSP Tags

Another important syntax element of JSP are tags. JSP tags do not use `<%`, but just the `<` character. A JSP tag is somewhat like an HTML tag. JSP tags can have a "start tag", a "tag body" and an "end tag". The start and end tag both use the tag name, enclosed in `<` and `>` characters. The end starts with a `/` character after the `<` character. The tag names have an embedded colon character: in them, the part before the colon describes the type of the tag. For instance:

```
<some:tag> body  
</some:tag>
```

If the tag does not require a body, the start and end can be conveniently merged together, as

```
<some:tag/>
```

Here by closing the start tag with a `/>` instead of `>` character, we are ending the tag immediately, and without a body. (This syntax convention is the same as XML.) Tags can be of two types: loaded from an external tag library, or predefined tags. Predefined tags start with `jsp:` characters. For instance, `jsp:include` is a predefined tag that is used to include other pages.

We have already seen the `include` directive. `jsp:include` is similar. But instead of loading the text of the included file in the original file, it actually calls the included target at runtime (the way a browser would call the included target. In practice, this is actually a simulated request rather than a full round-trip between the browser and the server). Following is an example of `jsp:include` usage

```
<HTML>  
<BODY>  
Going to include hello.jsp...<BR>  
<jsp:include page="hello.jsp"/>  
</BODY>  
</HTML>
```

Try it and see what you get. Now change the "`jsp:include`" to "`jsp:forward`" and see what is the difference. These two predefined tags are frequently very useful.

*Exercise:* Write a JSP to do either a `forward` or an `include`, depending upon a boolean variable (hint: The concepts of mixing HTML and scriptlets work with JSP tags also!)

### **JSP Sessions**

On a typical web site, a visitor might visit several pages and perform several interactions. If you are programming the site, it is very helpful to be able to associate some data with each visitor. For this purpose, "`session`"s can be used in JSP.

A session is an object associated with a visitor. Data can be put in the session and retrieved from it, much like a Hashtable. A different set of data is kept for each visitor to the site.

Here is a set of pages that put a user's name in the session, and display it elsewhere. Try out installing and using these.

First we have a form, let us call it `GetName.html`

```
<HTML>
<BODY>
<FORM METHOD=POST ACTION="SaveName.jsp">
What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20>
<P><INPUT TYPE=SUBMIT>
</FORM>
</BODY> </HTML>
```

The target of the form is "SaveName.jsp", which saves the user's name in the session. Note the variable "`session`". This is another variable that is normally made available in JSPs, just like `out` and `request` variables. (In the `@page` directive, you can indicate that you do not need sessions, in which case the "session" variable will not be made available.) <%

```
String name = request.getParameter( "username" ); session.setAttribute( "theName",
name );
%>
<HTML>
<BODY>
<A HREF="NextPage.jsp">Continue</A>
</BODY>
</HTML>
The SaveName.jsp saves the user's name in the session, and puts a link to another page,
NextPage.jsp.
```

NextPage.jsp shows how to retrieve the saved name.

```
<HTML>
<BODY>
Hello, <%= session.getAttribute( "theName" ) %>
</BODY>
</HTML>
```

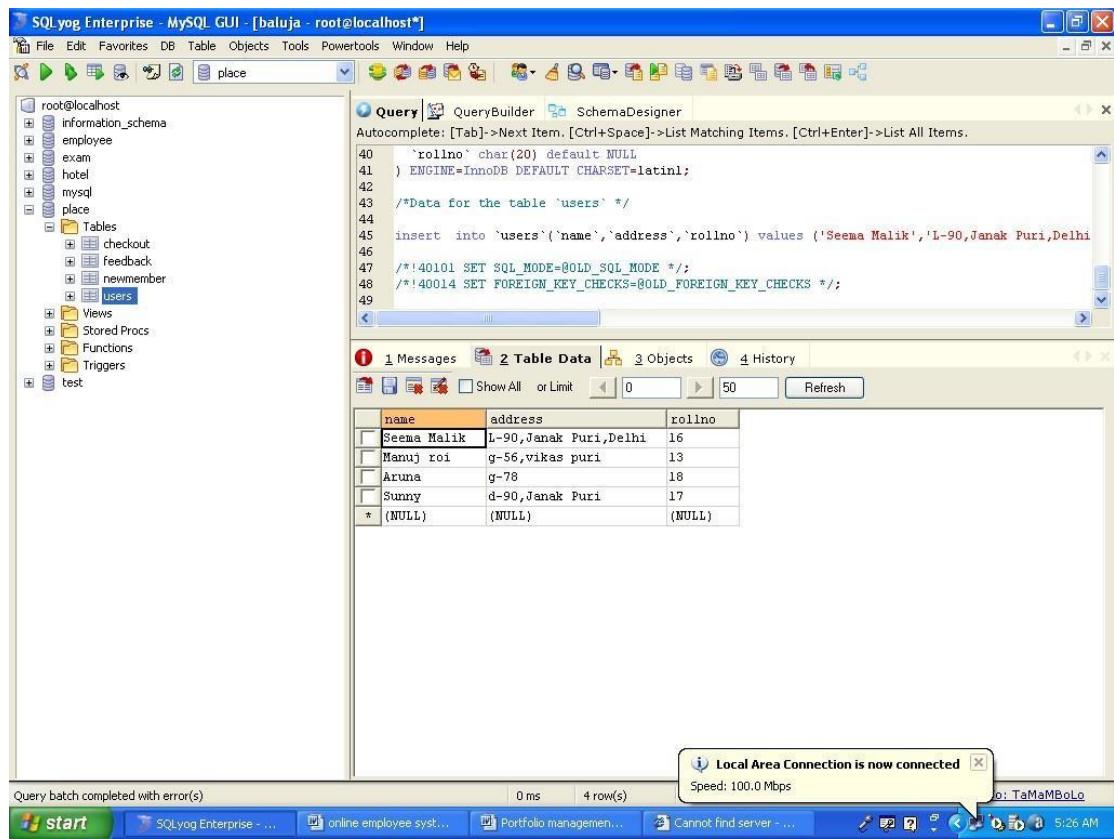
If you bring up two different browsers (not different windows of the same browser), or run two browsers from two different machines, you can put one name in one browser and another name in another browser, and both names will be kept track of.

The session is kept around until a timeout period. Then it is assumed the user is no longer visiting the site, and the session is discarded.

## **MY SQL**

### **Introduction**

My SQL is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA). MY SQL is probably the most accessible and the most documented enterprise database environment right now. This also means that you can learn it a little quicker than most other database environments on the market.



To start, you must have a computer that runs an appropriate operating system like Microsoft Windows >= XP Home Edition: that includes Windows XP Home Edition, Windows XP Professional, Windows 2000 Professional, or any version of Windows Server 2003. In this case, you must install MY SQL Yog.

### **What is SQL Used for:**

Using SQL one can create and maintain data manipulation objects such as table, views, sequence etc. These data manipulation objects will be created and stored on the server's hard disk drive, in a tablespace, to which the user has been assigned.

Once these data manipulation objects are created, they are used extensively in commercial applications.

### **DML, DCL, DDL:**

In addition to the creation of data manipulation objects, the actual manipulation of data within these objects is done using SQL.

The SQL sentences that are used to create these objects are called DDL's or Data Definition Language. The SQL sentences used to manipulate data within these objects are called DML's or Data Manipulation Language. The SQL sentences, which are used to control the behavior of these objects, are called DCL's or Data Control Language.

## DATA TYPE

### Built-in data types

In My Sql, each object (such as column, variable, or parameter) has a related data type, which is an attribute that specifies the type of data that the object can hold. My SQL ships with 27 built-in (system) data types. They are:

Data Types	Description
bigint	Integer data from -2^63 through 2^63-1
int	Integer data from -2^31 through 2^31 – 1
smallint	Integer data from -2^15 through 2^15 – 1
tinyint	Integer data from 0 through 255
bit	Integer data with either a 1 or 0 value
decimal	Fixed precision and scale numeric data from -10^38 +1 through 10^38 -1
numeric	Fixed precision and scale numeric data from -10^38 +1 through 10^38 -1
money	Monetary data values from -2^63 through 2^63 - 1
smallmoney	Monetary data values from -214,748.3648 through +214,748.3647
float	Floating precision number data from -1.79E + 308 through 1.79E + 308
real	Floating precision number data from -3.40E + 38 through 3.40E + 38

datetime	Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of 3.33 milliseconds
smalldatetime	Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute
char	Fixed-length character data with a maximum length of 8,000 characters
varchar	Variable-length data with a maximum of 8,000 characters
text	Variable-length data with a maximum length of $2^{31} - 1$ characters
nchar	Fixed-length Unicode data with a maximum length of 4,000 characters
nvarchar	Variable-length Unicode data with a maximum length of 4,000 characters
ntext	Variable-length Unicode data with a maximum length of $2^{30} - 1$ characters
binary	Fixed-length binary data with a maximum length of 8,000 bytes
varbinary	Variable-length binary data with a maximum length of 8,000 bytes
image	Variable-length binary data with a maximum length of $2^{31} - 1$ bytes
cursor	A reference to a cursor
sql_variant	A data type that stores values of various data types,

	except text, ntext, timestamp, and sql_variant
table	A special data type used to store a result set for later processing
timestamp	A database-wide unique number that gets updated every time a row gets updated
uniqueidentifier	A globally unique identifier
Some of these data types (bigint, sql_variant, and table) are only available in MySQL 8, while some were supported under the previous MySQL r versions.	

## User-defined data types

MySQL supports user-defined data types too. User-defined data types provide a mechanism for applying a name to a data type that is more descriptive of the types of values to be held in the object. Using user-defined data type can make it easier for a programmer or database administrator to understand the intended use of any object defined with the data type. The user-defined data types are based on the system data types and can be used to predefine several attributes of a column, such as its data type, length, and whether it supports NULL values. To create a user-defined data type, you can use the `sp_addtype` system stored procedure or you could add one using the Enterprise Manager. When you create a user-defined data type, you should specify the following three properties:

Data type's name.

Built-in data type upon which the new data type is based.

Whether it can contain NULL values. The following example creates a user-defined data type based on money data type named cursale that cannot be NULL:

```
EXEC sp_addtype cursale, money, 'NOT NULL'GO
```

Both system and user-defined data types are used to enforce data integrity. It is very important that we put forth a lot of effort while designing tables: the better you design your

tables, the more time you can work without any performance problems. In an ideal case, you never will update the structure of your tables.

## **ENVIRONMENT / OPERATING SYSTEM**

The age of standalone computing is long gone and it is now a world of networks. In this context, it is imperative for users to understand issues involved in working in a multi-user environment. Windows NT today is the most sought after Multi-user Operating System in the world. This is a 32-bit, powerful, user friendly and robust network operating system from Microsoft.

Windows NT is a big step up from Windows 95 both in terms of price and in terms of capabilities. NT Server is a well-built, fully functional, multi-user, multitasking operating system.

The release of NT Server 4 adds the popular Windows 8 user interface, the Internet Information Server (IIS) – capable of running Web, FTP, and Gopher services - a search engine called Index Server, the FrontPage HTML authoring package, and the Internet Explorer, all of which make Windows NT server a tough act to beat as a robust, capable web server.

The suggested Operating System for the server computer is Windows NT Server and for client computer is Windows 8.

## **2. SYSTEM STUDY**

### **2.1 SYSTEM DEVELOPMENT LIFE CYCLE**

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject System Analysis and Design (SAD) mainly deals with the software development activities.

#### **Defining a System**

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demands that some output is produced as a result of processing the suitable inputs.

## **System Life Cycle**

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

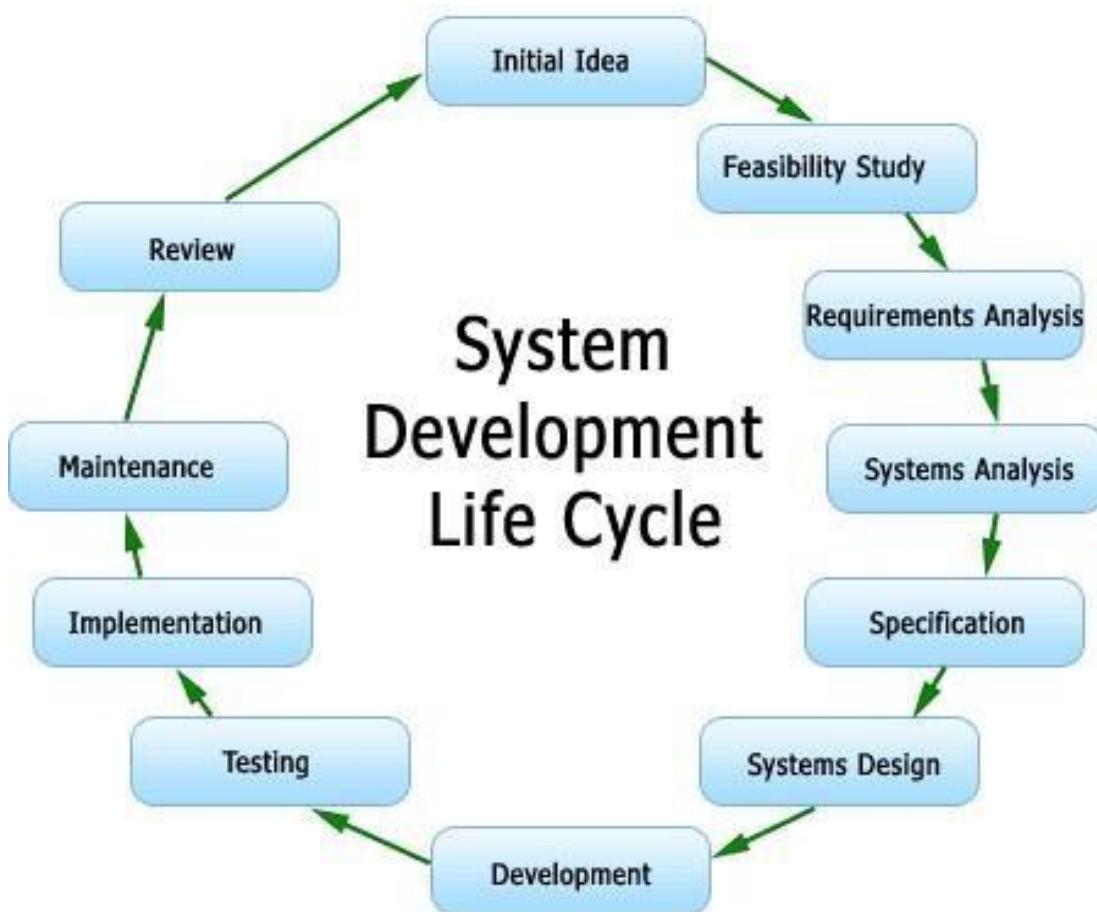
System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

### **Following are the different phases of software development cycle:**

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing

- Implementation
- Maintenance

**The Different Phases Of Software Development Life Cycle Are  
Shown Below.**



**FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED**  
**2.2 FEASIBILITY STUDY**

The basic premise of system analysis is being done here. The primary goal of the system analysis stage is to identify problems and determine how they can be solved with the computer system. In formal SDLC methodologies, the first step in system analysis is

feasibility study. A feasibility study is the quick examination of the problems, goals, expected cost of the system. The objective is to determine whether the problem can reasonably solved with a computer system. In some cases, may be there is a better alternative, or perhaps is simply short term annoyance and will gradually disappear. In other cases, the problem may turn out to be more complex than was thought and involves users across the company. Also, some problems may not be solvable with today's technology. It might be better to wait for better technology. In any case, you need to determine the scope of the project to gain the better idea of cost, benefits, and objectives.

The feasibility study is typically written so that non-programmers can easily understand it. It is used to "sell" the to the upper management and as a starting point for the next step. Additionally, it is used as a reference to keep the project on track, and to evaluate the progress of project team. Is the project cost effective or there is a cheaper solution? Will the proposed system improve the operation of the bank; will complicating factors prevent it from achieving its goals? Does the technology exist and does the firm have the staff to make the technology work?

When the proposal is determined to be feasible, the team leaders are appointed and a plan and schedule are created. The schedule contains a detailed listing of what parts of the project are completed at each time. Of course, it extremely difficult to estimate the true cost and completion dates. Nonetheless, the schedule is an important tool to evaluate the status of the project and the progress of the tea m.

---

### **Steps in feasibility Analysis are:**

1. Identify deficiency by pinpointing, Missing functions, unsatisfactory performance, Excessive cost of operations.
2. Set goals to remove these deficiencies.
3. Goals must be quantified, realizable within the constraints of an organization, broken down into sub goals agreeable to all concerned.
4. Set goals not only to remove deficiencies but also to effectively meet competition.  
For instance, goals must be based on what competitors do.

## **2.3 ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost / benefit analysis; in this procedure we determine the benefits and savings that are expected from a proposed system and compare them with costs. We found the benefits outweigh the costs; we take a decision to design and implement the new proposed system.

During the feasibility phase, broad alternatives solutions are examined. For each alternate solution the cost and benefits have to be examined before designing one of the alternative.

Broad solutions will consist of:

1. Specifications of information to be made available by the system.
2. Description of what will be done manually and what the computer will do.
3. Specification of new computing equipment needed or specification of expansion of an existing computer.

### **➤ COST AND BENEFIT ANALYSIS**

Developing an IT application is an investment. Since after developing that application it provided the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks because in some cases an estimate can be wrong and the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected cost.

In performing cost and benefit analysis it is important to identify cost and benefits factors.

Cost and benefits can be categorized into the following categories:

- i. **Development cost** - Development costs is the cost that are incurred during the development of the system. It is one-time investment.

- ii. **Operating cost** - Operating cost are the expenses required for the day to-day running of the system. As, operating cost are wages, supplies and overheads.
- iii. **Hardware/Software cost** - It includes the cost of purchasing or leasing of computers and its peripherals. Software costs involves required software cost.
- iv. **Personnel cost** - It is the money spent on the people involved in the development of the system.
- v. **Facility cost** - Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lighting, and air-conditioning.
- vi. **Supply cost** - These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and others.

## ➤ BENEFITS

We can define benefits as

Profit or Benefit = Income - Cost Benefits can be accrued by:

- Increasing income or
- Decreasing costs or
- Both

My proposed project to **The Perfect Trainer GYM Management System** does everything those 3 - 4 employees or accountants are currently doing on paperwork except on a computer. Due to this factor, if the type organization goes ahead with my project, they would not need any personnel, and their costs of sustaining the organization go down radically, the software itself requires minimal memory to run as files are stored in a very defragmented manner and can easily be moved around as well as zipped, to preserve even more space. Hence, funds spent on storage, are almost trifling, will not pose a problem in the future while operating my project.

All of this comes at a very low price of a computer and my software, which, if we consider all the benefits type organization will be getting out of it, is completely woo\* it.

Let's take the costs required to assemble and run my project

ITEM	COST (Rs.)
Computer	40,000 (depend upon the configuration)
Laser Printer	7,000
Scanner	4,000
Project cost	4,000 (approx.)
Total	55,000

We should know that we need a person who can work with this software's and so his training will cost around Rs. 4,000. Here we see that the total price to get one computer up and running with the database management system is close to Rs.50, 000 as we don't really need a very high performance computer as this project is entirely text based and requires only minimal processor speeds for computing. If the herbal store wants to have around 2 computers, then we have the total costs being:

$$55,000 * 2 = \text{Rs.} 1,10,000$$

This would mean that the Herbal Product would need around 1 lakh to run my system successfully and resourcefully. Herbal Product authorities are planning to reduce the staff, suppose if they remove one employee. Salary of one employee being around Rs. 5,000. In one-year annual salary of employee is equal to Rs. 60,000 and salary to three employees is Rs. 1,80,000 for one year. From this analysis it can be seen that whatever money the consultants or hospitals will invest on making their system a computerized system will be recovered and half years of time (approx.). This analysis shows that how management is benefited by computerized system.

## **BENEFITS:**

1. Fast and easy access to all the procedures and functions.
2. No need for large storage spaces sized of rooms for storing the cabinets because all the information about the members and other details is saved in the computer's hard disks.
3. High level of security and authentication of each and every user.
4. Reliability is increased, as backups of files, and records can be made and saved in various different locations and information will be highly secure, unlike in file cabinets where entries can easily be ripped or tampered with by users.
5. The reception/front office will look much neater and cleaner the environment they need, as there won't be any cupboards or drawers which make the organization overcrowded.
6. There will be no longer the need for all the paper work required to make timely reports lists or other lists as the program generates them at any time at a very quick pace.

## **2.4 TECHNICAL FEASIBILITY**

Today, very little is technically impossible. Consequently, technical feasibility looks at what is practical and reasonable. Technical feasibility addresses three major issues:

1. Is the proposed technology or solution practical?
2. Do we currently possess the necessary technology?
3. Do we possess the necessary technical expertise, and is the schedule reasonable?

### **Is the Proposed Technology or Solution Practical?**

The technology for any defined solution is normally available. The question whether that technology is mature enough to be easily applied to our problems. Some firms like to use state-of-the-art technology, but most firms prefer to use mature and proven technology. A mature technology has a larger customer base for obtaining advice concerning problems and improvements.

### **Do We Currently Possess the Necessary Technology?**

Assuming the solution's required technology is practical, we must next ask ourselves, is the technology available in our information systems shop? If the technology is available, we must ask if we have the capacity. For instance, will our current printer be able to handle the new reports and forms required of a new system?

If the answer to any of these questions is no, then we must ask ourselves, Can we get this technology? The technology may be practical and available,

and, yes, we need it. But we simply may not be able to afford it at this time. Although this argument borders on economic feasibility, it is truly technical feasibility. If we can't afford the technology, then the alternative that requires the technology is not practical and is technically infeasible!

### **We Possess the Necessary Technical Expertise, and Is the Schedule Reasonable?**

This consideration of technical feasibility is often forgotten during feasibility analysis. We may have the technology, but that doesn't mean we have the skills required to properly apply that technology. For instance, we may have a database management systems (DBMS). However, the analysis and programmers available for the project may not know that DBMS well enough to properly apply it. True, all information systems professionals can learn new technologies. However, that learning curve will impact the technical feasibility of the project; specifically, it will impact the schedule.

As mentioned earlier, the current operational state of consultants or hospitals is very primitive as all storage is done on hand written database. These files are then placed in drawers or cabinets and tagged in a sorted order. The offices contain over a large number of drawers and cabinets. Each cabinet takes a large space. Other than this it uses many king of papers to calculate and maintain different account works.

The software & hardware to run my project, and is its usage is given in the table below:

## **2.5 OPERATIONAL FEASIBILITY**

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

- o What changes will be brought with the system? What organizational structures are disturbed?
- o 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?

Generally, project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

For operational feasibility study we appointed a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and are skilled in system analysis and design process.

## **2.6 FEASIBILITY REPORT**

After studying the feasibility of the project we came to the following points, these results may change according to further analysis and design.

**PROJECT NAME:** The Perfect Trainer GYM MANAGEMENT SYSTEM

**EXPECTED BENEFITS:**

- Reduce the number of employee.
- Save money.
- Increase the efficiency of workers.
- Reduce the response time.
- Improve the service quality.
- Reduce the bulk of paper work.
- Reduce the chance of error by human.
- Increase the accuracy in result.

### **3.SYSTEM ANALYSIS**

#### **3.1 IMPORTANCE OF COMPUTERIZED The Perfect Trainer GYM MANAGEMENT SYSTEM**

There are several attributes in which the computer based information works. Broadly the working of computer system is divided into two main groups:

- ◆ Transaction System
- ◆ Decision Support System

##### **Transaction System:**

A transaction is a record of some well-defined single and usually small occurrence in a system. Transactions are input into the computer to update the database files. It checks the entering data for its accuracy. This means that numeric data appears in numeric field and character data in character field. Once all the checks are made, transaction is used to update the database. Transaction can be inputted in on-line mode or batch mode. In online mode, transactions are entered and updated into the database almost instantaneously. In batch mode, transactions are collected into batches, which may be held for a while and inputted later.

##### **Decision Support System:**

It assists the user to make analytical decision. It shows the various data in organized way called analysis. This analysis can be made to syrdy preferences and help in making decisions.

Computer system works out best with record maintenance. It will tell you which customer would get how much pending/reports statements. It will also help to search the information about a particular person by simply entering his telephone number. User can store information as per requirement, which can be used for comparison with other reports.

## **3.2 PRINCIPLES OF SYSTEM ANALYSIS**

### **Principles:**

1. Understand the problem before you begin to create the analysis model.
2. Develop prototypes that enable a user to understand how human machine interaction will occur.
3. Record the origin of and the reason for every requirement.
4. Use multiple views of requirements like building data, function and behavioural models.
5. Work to eliminate ambiguity.

### **A Complete Structure:**

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in news sites, but utmost care has been taken to make the system efficient and user friendly.

For the optimum use of practical time it is necessary that every session is planned.

Planning of this project will include the following things:

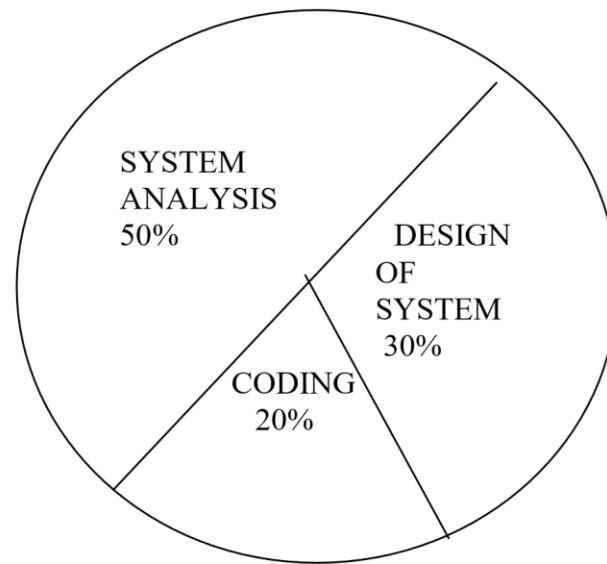
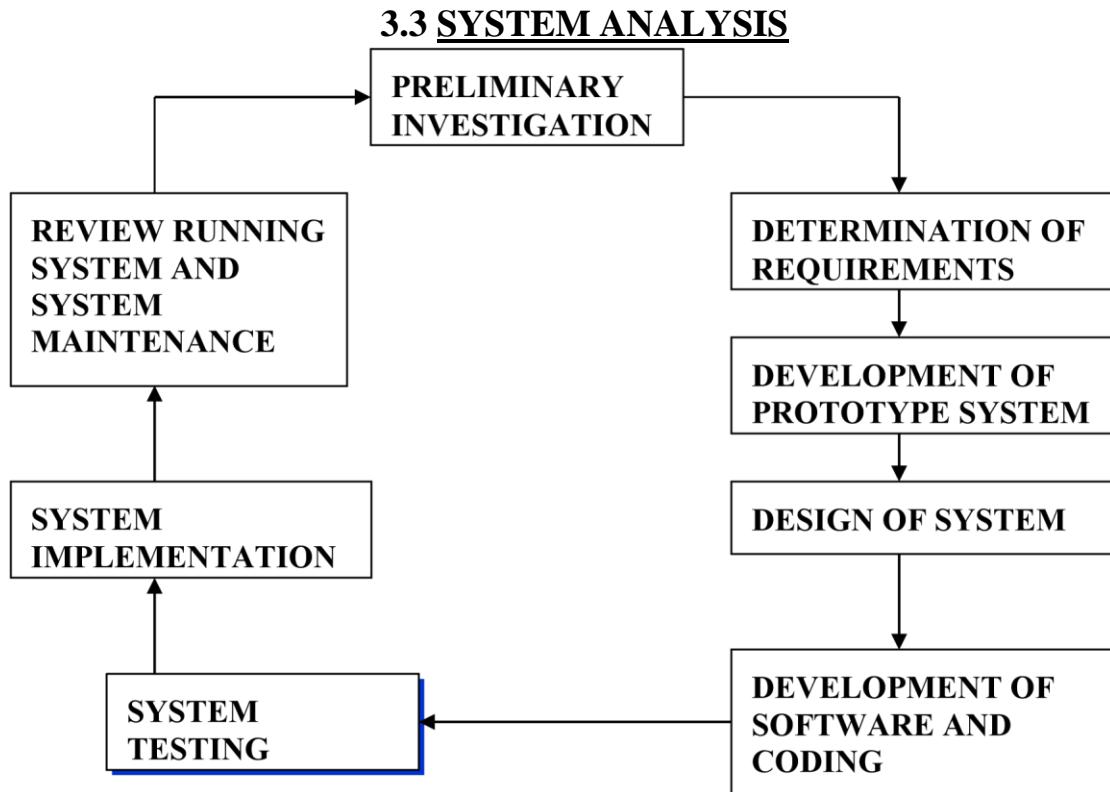
- Topic Understanding.
- Modular Break – Up of the System • Processor Logic for Each Module.
- Database Requirements.     **Topic Understanding:**

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as the project was allocated to me, I carefully went through the project to identify the requirements of the project.

### **Modular Break –Up of the System:**

- Identify The Various Modules in The System.
- List Them in The Right Hierarchy.

- Identify Their Priority of Development
- Description Of The Modules:



**FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED**

A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system, it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- What type of information is required?
- What are the constraints on the investigation?
- What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system, the total system was designed and can be described as under:

The three major parts of the system are:

➤ **Providing Information:**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide booking details, stock, sales with quick update to latest modifications in the records. This thing is not at all possible in

printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as a information provider for **The Perfect Trainer GYM Management System** administrator.

**Alert when available:** Through the survey it was clearly that there is a need to device an alternative way for providing alert facility to the user. Sometimes the product which customer demand is not available at that moment, user can register demand of customer and when its available, system gives an alert to the user that customer had registered a customer request with the same match.

**Constraints:** After the objectives were clear during the analysis phase, it was essential to understand the constraints in order to plan and avoid problems arising during detailed analysis.

**Technology** - the customer may be committed to a particular hardware or software solution. The software required in this case is: compete Java developer kit, Microsoft windows environment for MS - access.

**Budget** - if budget is a real constraint, the budget of the new system proposed would be constantly compared with that of the existing system or any Alternatives solution. In this case during the economic feasibility study it has been clearly proved that the new system is definitely more feasible than the alternative solution possible. Organization must implement a system which saves the effort, also its provide an easy method for customer who investigate each detail itself.

**Scope** - what is the area under investigation in this project? What are the boundaries of the system? What is the extent of possible usage of the new system?

More and more people are now having access to organization and watch independently Details of new upcoming stock. Hence the scope is constantly increasing. However, its

usage can be increased many folds with a little investment from the organization side by implanting touch screen computer kiosks at various convenient positions at the service station.

**Environmental Analysis:**

The external entities for an organization are its Supplier's customers or any individual.

### **3.4 METHODS USED FOR GATHERING INFORMATION**

The methods used for gathering information about the existing information system are as followed.

- (a) Review of records.
- (b) Observation of the functioning system.
- (c) Interviews.
- (d) Questionnaires.

In order to create an informative and practical system, a system analyst would have to have some kind of way to view the current system. Receiving feedback on what can be done to improve the current system, and how much the current system is acceptable to the users.

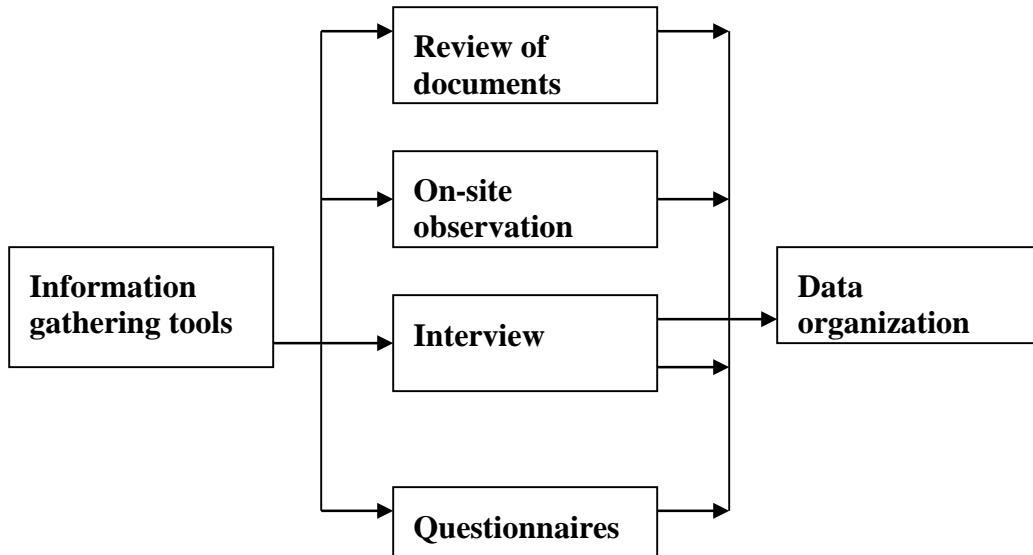
#### **1) Requirement analysis: -**

The main part of problem is to obtain a clear understanding of the needs of user and what exactly are desired from the software. It is used for specifying the requirement.

#### **Fact finding tools:**

After obtaining the background knowledge, I began to collect data on the existing system's output, input, and costs. The tools used in data collection / information gathering are:

- Review of the written Documents
- On-site Observation
- Interviews
- Questionnaires



### **Review of the written documents:**

In this phase we analysed all the documents like the day books, supply report, order generating forms, supply forms, account etc. All these things describe the format and functions of the current system included in most manuals are system requirement that help determine how III various objectives are met.

The form is one of the most important source through which! draw some conclusion like:

1. Who use the form(s)? How important are they to the user?
2. Do the forms include all the necessary information? What item should be added or deleted?
3. How readable and easy to follow is the form?
4. How does the information in the form help other user make better decision?
5. What other uses does the form offer the user area?

By analysing all the details, we draw a conclusion that what are the merit and De-merit of the current phase. Will the company contain all the back up of all the important document of not sales person contains all the information about the available vehicles or not? But

above all there are some problems with the onsite observations that one analyst must face during analysis like:

1. Take long time and get inefficient result
2. Attitude and motivation of subject cannot be readily observed
3. Observation are subject to error
4. In a complex situation it can be very time consuming

So for this we switched towards the other fact finding tools like interviews and questionnaires.

## **4. SYSTEM DESIGN**

### **4.1 SYSTEM DESIGN**

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process.

Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;

Determine into how many pieces to break; and

Identify the proper level of detail when design should stop and implementation should start.

Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain.

The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behaviour that interacts with each other in a fixed defined manner to produce some behaviours or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module.

The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. Reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

A design can be object-oriented or function-oriented. In function-oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object-oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function-oriented methods for design and describe one particular methodology the structured design methodology in some

detail. In a function- oriented design approach, a system is viewed as a transformation function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other.

Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other programmers. Generally, design specification uses textual structures, with design notation helping in understanding.

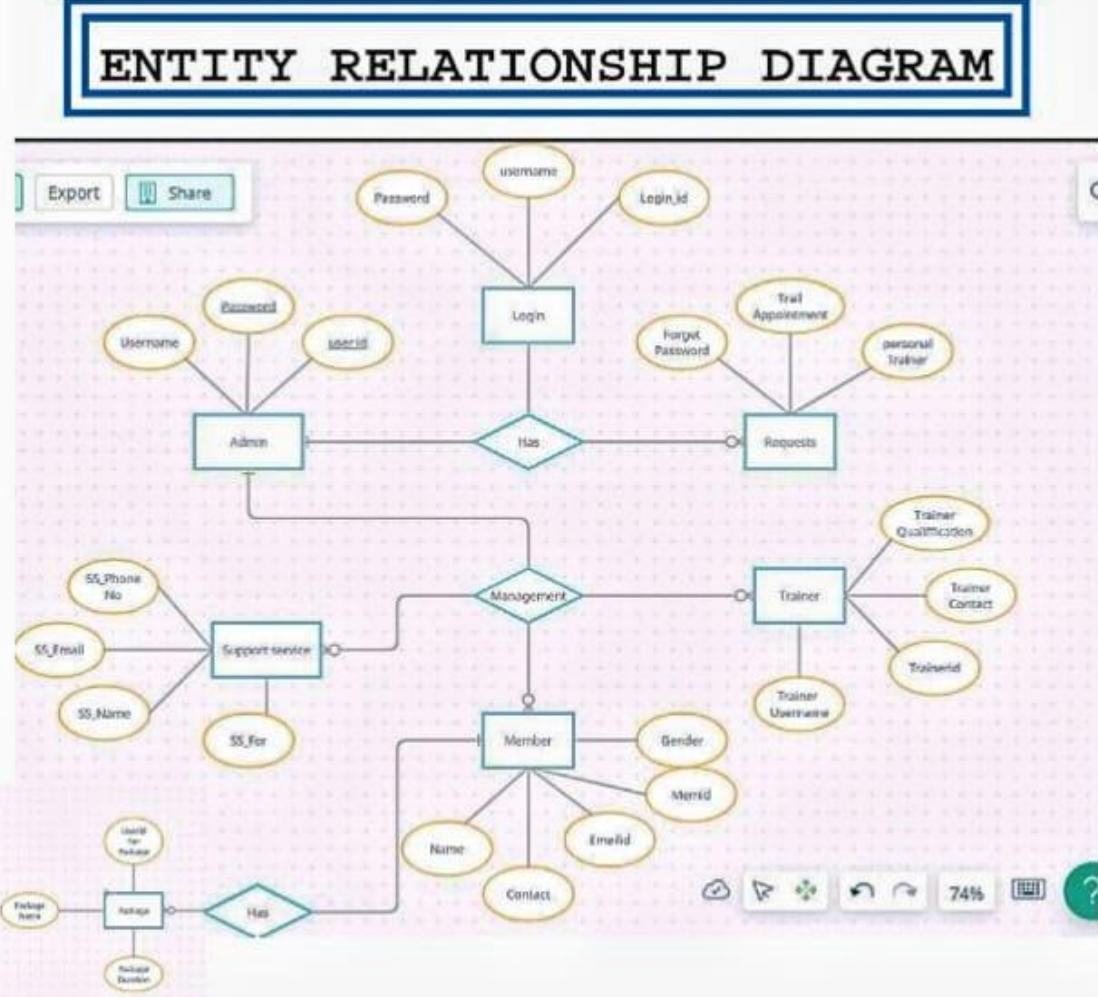
## **4.2 PHYSICAL DESIGN**

The design phase focuses on the detailed implementation of the system recommended in the feasibility. Emphasis is on translating performance specifications into design specifications. The design phase is a transition from user-oriented document to a programmer-oriented document.

### **4.2.1 Design Methodology:**

Design Methodology is a way to transform the "art" of system analysis and design into an "engineering - type" discipline. It explains the relationship amongst various modules and programs within the system. It standardizes the approach to analysis and design, simplifies design by segmentation, improves documentation and subsequent maintenance and enhancements.

# 4.3 ER Diagram



The following structured diagram can appropriately represent the relationship between various modules.

#### **4.2.2 Design Overview:**

In analysing the present system, a great deal of information was collected during the investigation and feasibility phases through list of problems and requirements, interview reports, questionnaires, onsite observations, manuals and determining potential solutions.

It is important to record this information in an unambiguous, concise manner which will be clear and accessible to others, and which can be used by other analysts and designers involved in developing the system. Structured techniques help us to record the information in this way, using diagrams and minimum amount of the text.

Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specification that are easily understandable to the user. The traditional approach of organizing data through flowcharts support future developments and simplify communication with the user but focus on the cost/benefit and feasibility analysis, project management, hardware and software selection, and personal considerations. In contrast, structured analysis considers new goals and structured tools for analysis, which provide the basis for design and implementation.

#### **4.2.3 Process Modelling:**

System design goes through two phases of development: logical and physical. Logical implementation represented by Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system it describes the input (source), outputs (destinations), data bases (data stores), and procedures (data flows) - all in the format that meets the user's requirements. The logical implementation of the whole project can be represented as under through Data Flow Diagrams (DFD).

### **4.4 DATA FLOW DIAGRAM**

Data flow diagrams are the most regularly utilized method for reporting the handling of the hopeful framework. As their name recommend they are a pictorial method for speaking to the stream of data into, around, and out of the framework. They are effortlessly justifiable and are less inclined to distortion than printed depiction. A complete arrangement of DFDs gives a reduced top - down representation of the framework, which makes it less demanding for the client and the investigator to conceive the framework all in all.

DFDs are built utilizing four noteworthy parts:

- **External elements** - speaks to the wellsprings of the data that enter the framework or the beneficiaries of the framework that leave the framework.

**for instance** - traveller is the typical beneficiary of data and supplier of data amid structure filling.

- **Data stores** - speak to the stores of the data inside the framework case: COMPUTER documents, databases or in the manual framework records, and so on data stores can not be connected specifically by data streams either to each other or to outer elements without an interceding procedure to change them.
- **Processes** - speak to exercises in which data is controlled by being put away or recovered or changed somehow.

Process names are by and large unambiguous and pass on however much importance as could reasonably be expected without being too long. Case: check data, obtained time plan and so on.

**Data flows** - speaks to the development of data between different parts.

# DATA FLOW DIAGRAM

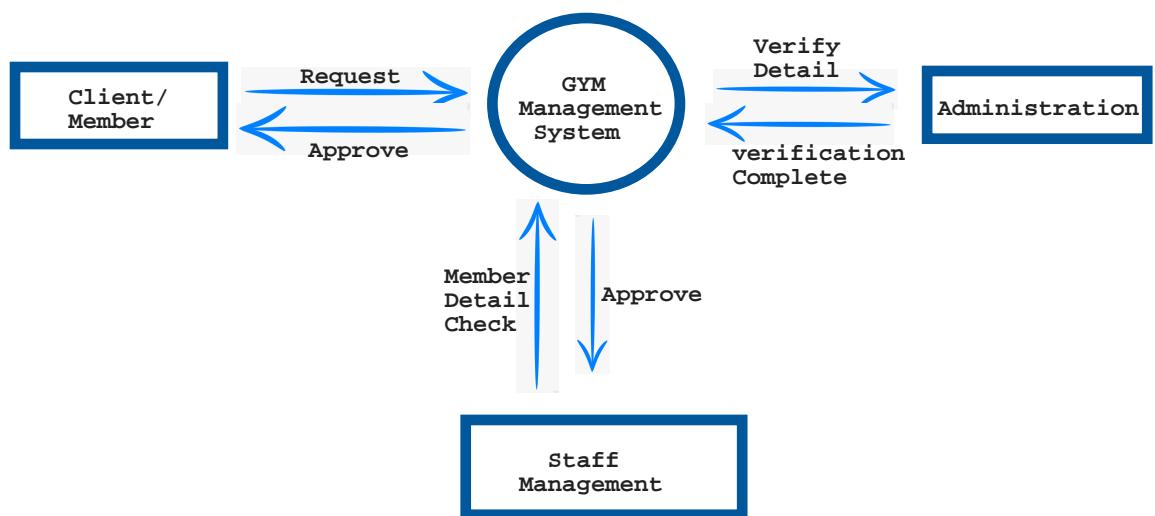


Figure:- DFD Level 0

## DATA FLOW DIAGRAM

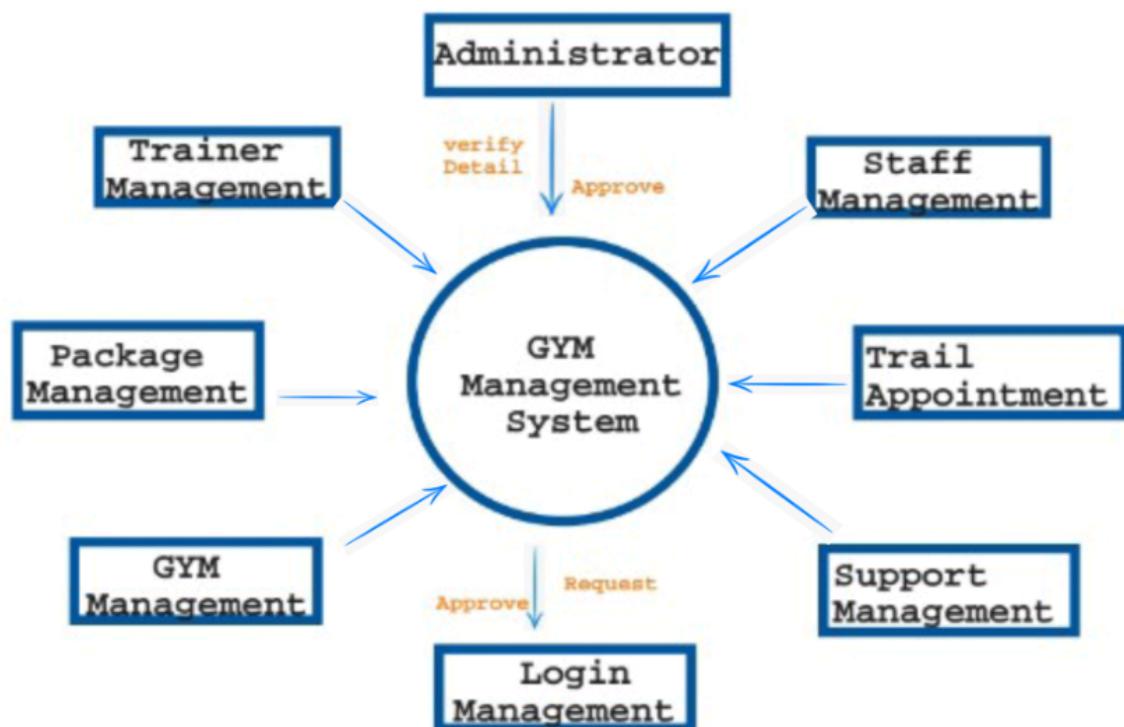


Figure:1 DFD LEVEL 1

## DATA FLOW DIAGRAM

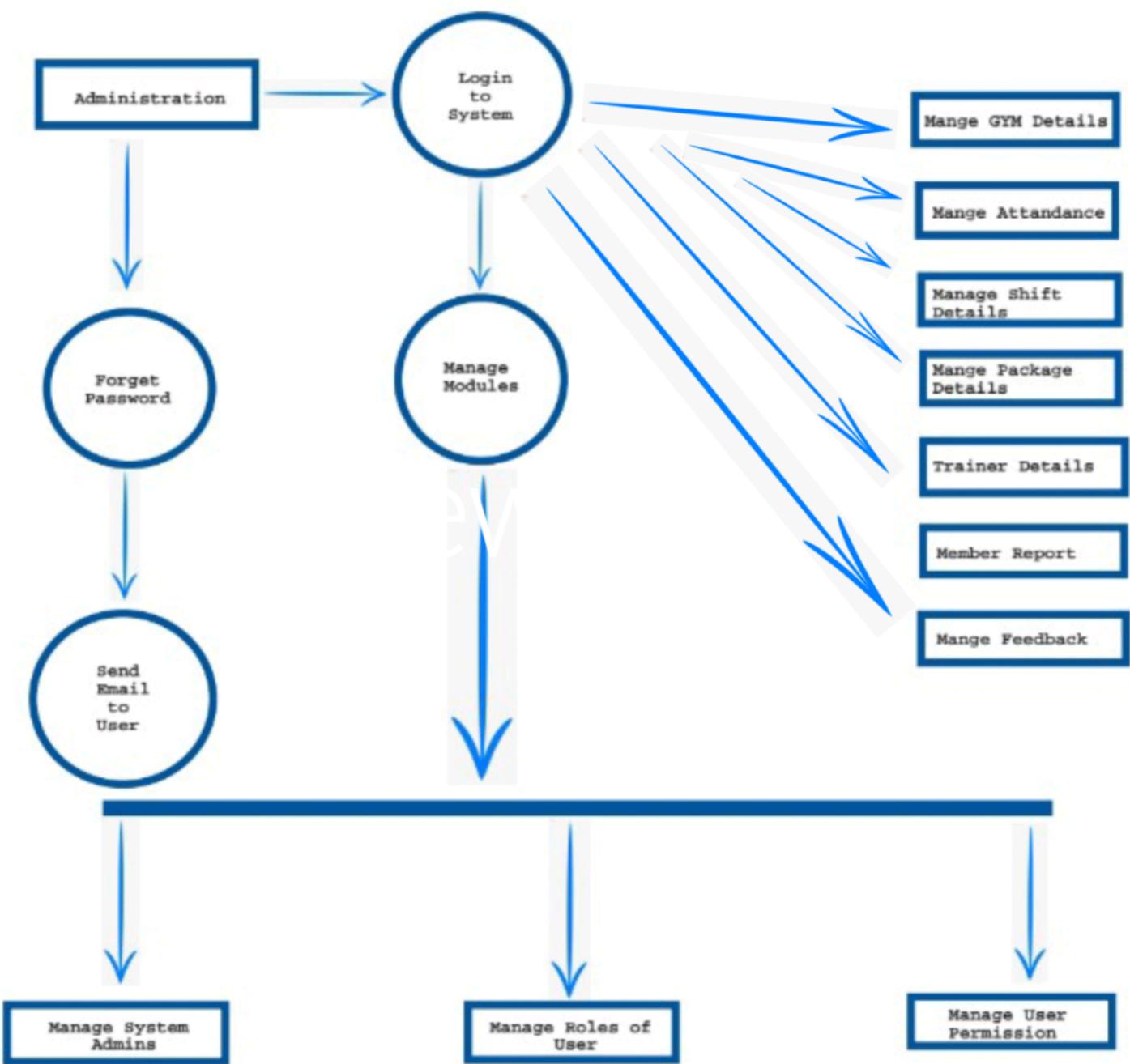


Figure: DFD Level 2

# DATABASE STRUCTURE

**Administrator**

S. No.	Field Name	Data Type	Description
1.	A_name	Varchar (30)	Admin Name
2.	A_id	Varchar (10)	admin unique id
3.	A_pass	Varchar (10)	admin password
4.	A_email	Varchar (30)	admin Email Id

**Trainer Detail**

S. No.	Field Name	Data Type	Description
1.	to_id	Varchar (15)	Trainer Id
2.	to_location	Varchar (20)	Trainer Gym Location
3.	to_name	Varchar (30)	Triner Name
4.	to_address	Varchar (50)	Trianer Adress
5.	to_cont_no.	Varchar (15)	Trainer Contact no
6.	to_email	Varchar (20)	Trainer Email

**Member Deatil**

S. No.	Field Name	Data Type	Description
1.	Mem_Id	Varchar (15)	Member Id
2.	Mem_Name	Varchar (20)	Member Name
3.	Mem_Address	Varchar (30)	Member Address
4.	Mem_GYM Location	Varchar (50)	GYM Address
5.	Mem_Package	Varchar (15)	Member Package
6.	Mem_shift	Varchar (20)	Member Shift Time

#### **Trail Appoinment**

S. No.	Field Name	Data Type	Description
1.	Name	Varchar (10)	Name of
2.	Contact No	Varchar (10)	Contact No of
3.	Email Id	Varchar (20)	Email Id of
4.	Feedback	Varchar (100)	Feedback of

#### **Feedback Module**

S. No.	Field Name	Data Type	Description
1.	Name	Varchar (10)	Name of Traveller
2.	Contact No	Varchar (10)	Contact No of Traveller
3.	Email Id	Varchar (20)	Email Id of Traveller
4.	Feedback	Varchar (100)	Feedback of Traveller

# **MODULES OF THE PROJECT**

This product is divided into number of modules. So that it can be easily manage.

## **USER SIDE**

Registration:-This module will help to register any customer. He/she has to provide all the information mention in the list. This module can be open by any one.

Login:-we are login in this software to use benefit the project and no one use your account before login we should the registration in valid e-mail id and password and name.

Search:-This module are use easily to search the product, this module save the user's time.

Payment:-This module user can order the product .user can choose the mode of payment with this module. Like credit card, debit card, net banking and cash on delivery. This mode to pay the product payment.

Feedback:-This module will take feedback of the customer who are using this module he/she will give his/her suggestion and organization so that organization can take proper action to improve service.

## **ADMIN SIDE**

Admin:-This module is only for administrator. One can only open this module by giving proper user password and username. This module will help to change product information like product name, category, price and any other information.

Manage Items:-The Administrator can access all data stored in database and also delete them but can't modify the some characteristics like(initial price , Description ,etc.)

Manage User: -The Administrator can access the data about user and delete them.

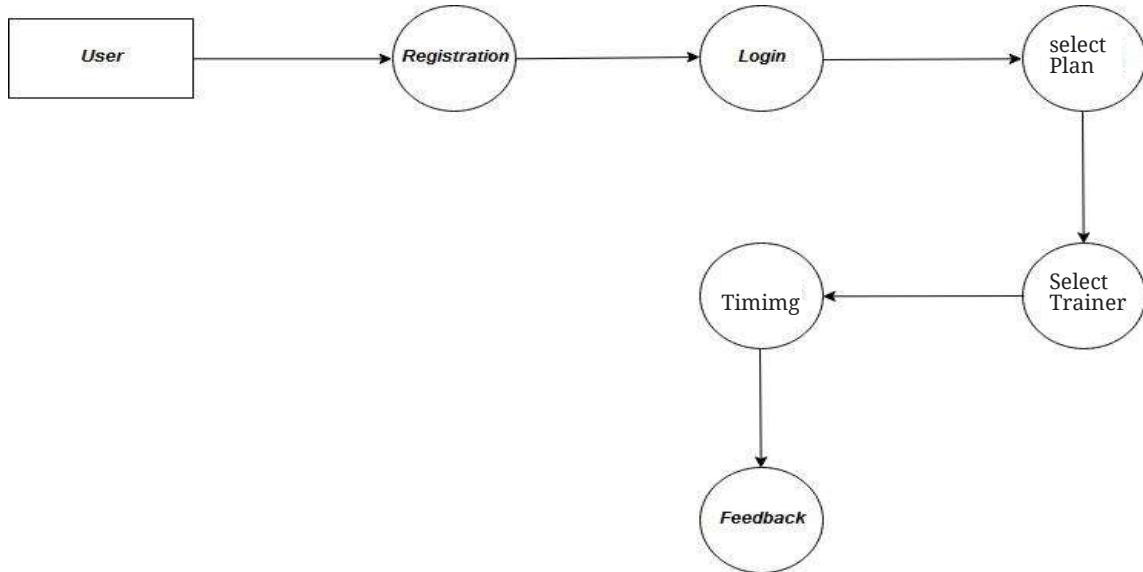
Manage Category:-The Administrator can access and modify all data about category stored in database ,add a new category and delete them . The Administrator can delete a category if any item is not associated with that category.

Feedback:-The administrator can view all feedbacks that user gives about application

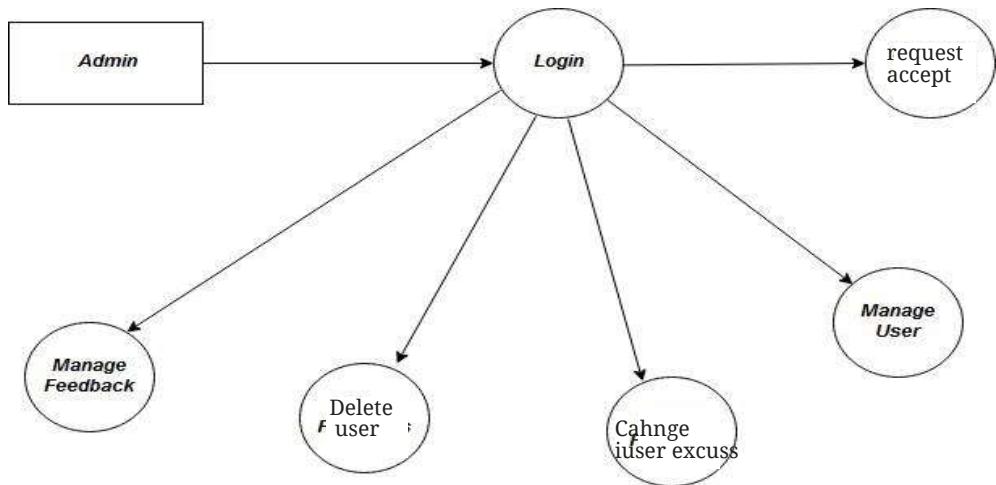
## 4.6 PROCESS LOGIC FOR EACH MODULE:

Anyone can open this project by just giving right user id and password. Then one can do general enquiry. But anyone want to the product to has to first registered itself. This is done for security reasons.

### PROCESS LOGIC FOR USER



### PROCESS LOGIC FOR ADMIN



# Feedback Module for Gym Management System

## Features

### 1. Feedback Collection:

- Members can submit feedback via a user-friendly interface.
- Categories for feedback:
  - Gym Facilities (Equipment, Cleanliness)
  - Trainer Performance
  - Customer Service
  - Overall Experience
- Star Rating System (1-5 stars) for quick reviews.

### 2. Feedback Management:

- Feedback submissions are stored in a centralized database.
- Filter feedback based on categories, star ratings, or dates.

### 3. Feedback Analytics:

- Summarize average ratings for each category.
- Highlight most common issues or suggestions using data visualization (e.g., pie charts or bar graphs).

### 4. Admin Responses:

- Admins can respond to feedback to acknowledge and resolve issues.
- Status tracking (e.g., "Pending," "Resolved").

### 5. Anonymous Feedback:

- Option for members to submit feedback anonymously to encourage honesty.

### 6. Email Notifications:

- Notify admins when new feedback is submitted.
  - Optional automated replies thanking members for their feedback.
-

### **Database Schema (Example)**

Table: Feedback

Field Name	Data Type	Description
Feedback_ID	INT (Primary Key)	Unique identifier for feedback.
Member_ID	INT (Foreign Key)	Member submitting the feedback.
Category	VARCHAR(50)	Feedback category (e.g., Facilities).
Rating	INT	Star rating (1-5).
Comments	TEXT	Detailed comments from the member.
Date_Submitted	DATETIME	Timestamp of submission.
Admin_Response	TEXT	Response from admin (if any).
Status	VARCHAR(20)	Status of feedback (Pending/Resolved).

---

### **->Sample User Interface**

#### **1. Member Feedback Form:**

- Dropdown menu for category selection.
- Star rating slider.
- Textbox for comments.
- Checkbox for submitting anonymously.
- Submit button.

#### **2. Admin Feedback Dashboard:**

- List of feedback with filters and sorting.
  - Visual representation of ratings and trends.
  - Option to reply to feedback directly from the dashboard.
-

# SYSTEM DEVELOPMENT

## LOGIN SOURCE:-

```
import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 *
 * @author Shan Wijenayaka
 */
public class LoginFunc {
    private static String logName;
    public void setLogName(String lg){
        logName = lg;
    }
    public String getLogName(){
        return logName;
    }
    public boolean checkLogin(String name, String password){
        Connection conn = new DBConnection().connect();
        String sql = "select * from login where name = ? and pword =? ";
        try{
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1,name);
            ps.setString(2, password);
            ResultSet rs = ps.executeQuery();

            if(rs.next()){
                //LogIn.this.setVisible(false);
                //JOptionPane.showMessageDialog(null, "welcome");
                System.out.println("correct uname pword");

                new Home().setVisible(true);

                setLogName(name);
                return true;
                //System.out.println("logger value from login clz :" +log);
            }
            else{
                JOptionPane.showMessageDialog(null,"Username or Password is incorrect");
                System.out.println("incorrect uname pword"+name+" "+password);
                return false;
            }
        } catch (SQLException ex) {
            System.out.println("sql exception in submit btn :" +ex);
            // Logger.getLogger(LogIn.class.getName()).log(Level.SEVERE, null, ex);
        }
        catch(HeadlessException ez){
            System.out.println("Error :" +ez);
        }
    }
    return false;
}
```

# GYM MANAGEMENT SYSTEM

User Name:

Password

Log in

## ADMINISTRATOR ADD MEMBER

ADD MEMBER SOURCE CODE:-

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JOptionPane;

/**
 *
 * @author Shan Wijenayaka
 */
public class AddMemberFunc {

    private static int lastid;

    Connection conn = new DBConnection().connect();
    void addToMemberDetails(String memName, String contactNo, String memAddress) {
        try {
            PreparedStatement ps;
            ps = conn.prepareStatement("insert into "
                    + "memberdetails" + "(name,contact_number,address,registered_on,registered_by) "
                    + "values(?,?,?,?,?)");
        }
    }
}
```

```

ps.setString(1, memName);
ps.setString(2, contactNo);
ps.setString(3, memAddress);

java.sql.Timestamp date = new java.sql.Timestamp(new java.util.Date().getTime());
ps.setTimestamp(4, date);

ps.setString(5, new LoginFunc().getLogName());
JOptionPane.showMessageDialog(null, memName+"'s information successfully entered to the database");

ps.executeUpdate();
new Home().setVisible(true);

} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, "error: "+ex);
}

// void addTOMoreMemDetails(String NIC, String emailAddress, String emContact, String gender, String remarks)

void addTOMoreMemDetails(String NIC, String emailAddress, String emContact,
String Gender, String Remarks, Double H, Double W, String MemberCat, String ReceiptNo, String payments) {
try {
    PreparedStatement ps;
    ps = conn.prepareStatement("insert into "
        + "morememberdetails" + "(id,nic,email,emcontact,gender,height,weight,bmi,remarks,lastUpdated,membership"
        + "values(?,?,?,?,?,?,?,?,?,?)");
    ps.setInt(1, getId());
    ps.setString(2, NIC);
    ps.setString(3, emailAddress);
    ps.setString(4, emContact);
    ps.setString(5, Gender);
    ps.setDouble(6, H);
    ps.setDouble(7, W);
    ps.setInt(8, getBMI(H, W));
    ps.setString(9, Remarks);
    java.sql.Timestamp date = new java.sql.Timestamp(new java.util.Date().getTime());
    ps.setTimestamp(10, date);
    ps.setString(11, MemberCat);
    ps.setString(12, ReceiptNo);
    ps.setString(13, payments);

    ps.executeUpdate();
    JOptionPane.showMessageDialog(null, "information successfully entered to the database");

} catch (SQLException ex) {
    Logger.getLogger(AddMemberFunc.class.getName()).log(Level.SEVERE, null, ex);
}
}

public int getId(){
try{

```

```

PreparedStatement st = conn.prepareStatement("select * from memberdetails where id");

ResultSet rs = st.executeQuery();
rs = st.executeQuery("SELECT MAX(id) AS id FROM memberdetails");
if(rs.next())
    lastid = rs.getInt("id");
System.out.println("last id from the function :" +lastid);

return lastid;
}catch(Exception e){
    JOptionPane.showMessageDialog(null,"problem at setting foriegn key: "+e );
    System.out.println(e);
}

return lastid;
}

public int getBMI(double h, double w){
    int b = (int) (w/(h*h));
    return b;
}

```

## Add New Member

Name: <input type="text"/>	Membership Category: <input type="text" value="Main workout area"/>
NIC: <input type="text"/>	Receipt No: <input type="text"/>
Contact Number: <input type="text"/>	Plan Time <input type="text" value="6 months"/>
Address(Residence): <input type="text"/>	Gender: <input type="text" value="Male"/>
Email: <input type="text"/>	Height: <input type="text"/> m
Emergency Contact No: <input type="text"/>	Weight: <input type="text"/> kg
<input type="button" value="Clear Text Fileds"/>	Remarks: <input type="text"/>
<input type="button" value="Add to the Database"/>	
<input type="button" value="Close"/>	

# TRAINER DETAILS

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.JOptionPane;

/**
 *
 * @author Shan Wijenayaka
 */
public class UpdateTraFunc {
    private static int lastid;
    private static String myID;

    public UpdateTraFunc(String MemID){
        myID = MemID;
    }

    Connection conn = new DBConnection().connect();
    void addToTrainerDetails(String memName, String contactNo, String memAddress) {
        try {
            //Update trainer_info SET name =?,idNo = ?,membershipCat =?,mNum = ?,hNum = ?,email=?,birthday=?,
            //age=?,gender=?,spouseName=?,address=?,height=?,weight=?
            //,bmi=?,eName=?,eRelation=?,emNum=?,eAddress=?,Remarks=?,fullPayment=? where membershipNo = ?"
            PreparedStatement ps = conn.prepareStatement("Update memberdetails SET name =?,contact_number =?",
                ,address =? where id = ?");

            ps.setString(1, memName);
            ps.setString(2, contactNo);
            ps.setString(3, memAddress);
            ps.setString(4,myID);
            ps.executeUpdate();

            JOptionPane.showMessageDialog(null, memName+"'s information successfully entered to the database");

        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "error: "+ex);
        }
    }

    // void addTOMoreMemDetails(String NIC, String emailAddress, String emContact, String gender, String
    remarks)
```

```

void addTOMoreTraDetails(String NIC, String emailAddress, String emContact,
    String Gender, String Remarks, Double H, Double W, String MemberCat, String ReceiptNo ,
    String payments) {
try {
    PreparedStatement ps;
    ps = conn.prepareStatement("Update moretrainerdetails SET nic =?,email =?,emcontact =?
    ,gender =?,height =?,weight =?,bmi =?,remarks = ?,lastUpdated = ?,membership_cat =?,
    Salary_plan =? where id = ?");

    ps.setString(1, NIC);
    ps.setString(2, emailAddress);
    ps.setString(3, emContact);
    ps.setString(4, Gender);
    ps.setDouble(5, H);
    ps.setDouble(6, W);
    ps.setInt(7, getBMI(H,W));
    ps.setString(8, Remarks);
    java.sql.Timestamp date = new java.sql.Timestamp(new java.util.Date().getTime());
    ps.setTimestamp(9, date);
    ps.setString(10, MemberCat);

    ps.setString(11, payments );
    ps.setString(12, myID );

    ps.executeUpdate();
    JOptionPane.showMessageDialog(null, "information successfully entered to the database");
}
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null,"error at updating more member details table: "+ex );
}
}

```

```

public int getId(){
try{
    PreparedStatement st = conn.prepareStatement("select * from memberdetails where id");

    ResultSet rs = st.executeQuery();
    rs = st.executeQuery("SELECT MAX(id) AS id FROM memberdetails");
    if(rs.next())
        lastid = rs.getInt("id");
    System.out.println("last id from the function :" +lastid);

    return lastid;
} catch(Exception e){
    JOptionPane.showMessageDialog(null,"problem at setting foreign key: "+e );
    System.out.println(e);
}

return lastid;
}

public int getBMI(double h, double w){
    int b = (int) (w/(h*h));
    return b;
}
}

```

# Trainer Details

Name: <input type="text"/>	Membership Category: <input type="text" value="Main workout area"/>
NIC: <input type="text"/>	Salary <input type="text" value="6 months"/>
Contact Number: <input type="text"/>	
Address(Residence): <input type="text"/>	
Email: <input type="text"/>	Gender: <input type="text" value="Male"/>
Emergency Contact No: <input type="text"/>	Height: <input type="text"/> m
<input type="button" value="Set Previously saved States"/>	Weight: <input type="text"/> kg
<input type="button" value="Clear Text Fields"/>	Remarks: <input type="text"/>
<input type="button" value="Add to the Database"/>	
<input type="button" value="Close"/>	

## DELETE MEMBER

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Member {
    private int id;
    private String name;
    // other member details

    public Member(int id, String name) {
        this.id = id;
        this.name = name;
    }

    // getters and setters for member details
    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }
}

class GymManagementSystem {
    private List<Member> members;

    public GymManagementSystem() {
        members = new ArrayList<>();
    }
}
```

```

public void addMember(Member member) {
    members.add(member);
}

public void deleteMember(int memberId) {
    for (int i = 0; i < members.size(); i++) {
        if (members.get(i).getId() == memberId) {
            members.remove(i);
            System.out.println("Member with ID " + memberId + " has been deleted.");
            return;
        }
    }
    System.out.println("Member with ID " + memberId + " not found.");
}

public void displayMembers() {
    System.out.println("---- Members ----");
    for (Member member : members) {
        System.out.println("ID: " + member.getId() + ", Name: " + member.getName());
    }
    System.out.println("-----");
}
}

public class GymManagementSystemApp {
    public static void main(String[] args) {
        GymManagementSystem gym = new GymManagementSystem();

        // Adding some sample members
        gym.addMember(new Member(1, "John Doe"));
        gym.addMember(new Member(2, "Jane Smith"));
        gym.addMember(new Member(3, "Mike Johnson"));

        // Display initial member list
        gym.displayMembers();

        // Delete a member
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the ID of the member to delete: ");
        int memberId = scanner.nextInt();
        gym.deleteMember(memberId);

        // Display updated member list
        gym.displayMembers();

        scanner.close();
    }
}

```

## Remove Member

```
import java.util.ArrayList;
import java.util.List;

class Member {
    private int id;
    private String name;
    // other member details

    public Member(int id, String name) {
        this.id = id;
        this.name = name;
    }

    // getters and setters for member details
    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }
}

class GymManagementSystem {
    private List<Member> members;

    public GymManagementSystem() {
        members = new ArrayList<>();
    }

    public void addMember(Member member) {
        members.add(member);
    }

    public void deleteMember(int memberId) {
        for (int i = 0; i < members.size(); i++) {
            if (members.get(i).getId() == memberId) {
                members.remove(i);
                System.out.println("Member with ID " + memberId + " has been deleted.");
                return;
            }
        }
        System.out.println("Member with ID " + memberId + " not found.");
    }

    public void displayMembers() {
        if (members.isEmpty()) {
            System.out.println("No members found.");
            return;
        }

        System.out.println("---- Members ----");
        for (Member member : members) {
            System.out.println("ID: " + member.getId() + ", Name: " + member.getName());
        }
        System.out.println("-----");
    }
}

public class GymManagementSystemApp {
    public static void main(String[] args) {
        GymManagementSystem gym = new GymManagementSystem();

        // Adding some sample members
        gym.addMember(new Member(1, "John Doe"));
        gym.addMember(new Member(2, "Jane Smith"));
        gym.addMember(new Member(3, "Mike Johnson"));

        // Display all members
        gym.displayMembers();
    }
}
```

# Gym Management System

[overview](#)[get member info](#)[Financial Details](#)

## Add New Member

### At a Glance...

Date

time

# DATABASE SQL QUERY

-- Create the members table  
CREATE TABLE members (  
 member\_id INT PRIMARY KEY,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 email VARCHAR(100),  
 phone\_number VARCHAR(20),  
 membership\_type VARCHAR(50)  
);

-- Create the trainers table  
CREATE TABLE trainers (  
 trainer\_id INT PRIMARY KEY,  
 first\_name VARCHAR(50),  
 last\_name VARCHAR(50),  
 email VARCHAR(100),  
 phone\_number VARCHAR(20)  
);

-- Create the classes table  
CREATE TABLE classes (  
 class\_id INT PRIMARY KEY,  
 class\_name VARCHAR(100),  
 trainer\_id INT,  
 class\_time DATETIME,  
 class\_capacity INT,  
 FOREIGN KEY (trainer\_id) REFERENCES trainers(trainer\_id)  
);

-- Create the bookings table  
CREATE TABLE bookings (  
 booking\_id INT PRIMARY KEY,  
 member\_id INT,  
 class\_id INT,  
 booking\_time DATETIME,  
 FOREIGN KEY (member\_id) REFERENCES members(member\_id),  
 FOREIGN KEY (class\_id) REFERENCES classes(class\_id)  
);

-- Insert sample data into members table  
INSERT INTO members (member\_id, first\_name, last\_name, email, phone\_number, membership\_type)  
VALUES (1, 'John', 'Doe', 'john.doe@example.com', '1234567890', 'Premium'),  
 (2, 'Jane', 'Smith', 'jane.smith@example.com', '9876543210', 'Basic');

-- Insert sample data into trainers table  
INSERT INTO trainers (trainer\_id, first\_name, last\_name, email, phone\_number)  
VALUES (1, 'Mike', 'Johnson', 'mike.johnson@example.com', '5555555555'),  
 (2, 'Emily', 'Davis', 'emily.davis@example.com', '4444444444');

-- Insert sample data into classes table  
INSERT INTO classes (class\_id, class\_name, trainer\_id, class\_time, class\_capacity)  
VALUES (1, 'Yoga', 1, '2023-06-01 10:00:00', 20),  
 (2, 'Zumba', 2, '2023-06-02 15:30:00', 15);

-- Insert sample data into bookings table  
INSERT INTO bookings (booking\_id, member\_id, class\_id, booking\_time)  
VALUES (1, 1, 1, '2023-05-31 09:00:00'),  
 (2, 2, 2, '2023-05-31 13:00:00');

## **5.2 CODE EFFICIENCY**

Software Testing is an empirical investigation conducted to provide stakeholders with of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. information about the quality of the product or service under test , with respect to he context in which it is intended to operate. This includes, but is not limited to, the process

## **5.3 OPTIMIZATION OF CODE**

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in a optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small.

## **6. TESTING**

### **6.1 TESTING**

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.<sup>[11]</sup> The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

#### **Defects and failures**

Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirements 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.

Software faults occur through the following process. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure.<sup>[12]</sup> Not all defects will necessarily result in failures. For example, defects in dead code will never result in failures. A defect can turn into a failure when the environment is changed. Examples of these changes in environment include the software being run on a new hardware platform, alterations in source data or interacting with different software.<sup>[12]</sup> A single defect may result in a wide range of failure symptoms.

## **Compatibility**

A frequent cause of software failure is compatibility with another application, a new operating system, or, increasingly, web browser version. In the case of lack of backward compatibility, this can occur (for example...) because the programmers have only considered coding their programs for, or testing the software upon, "the *latest* version of" this-or-that operating system. The unintended consequence of this fact is that: their latest work might not be fully compatible with earlier mixtures of software/hardware, or it might not be fully compatible with *another* important operating system. In any case, these differences, whatever they might be, may have resulted in (unintended...) software failures, as witnessed by some significant population of computer users.

This could be considered a "prevention oriented strategy" that fits well with the latest testing phase suggested by Dave Gelperin and William C. Hetzel, as cited below [13].

## **Input combinations and preconditions**

A very fundamental problem with software testing is that testing under *all* combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More significantly, nonfunctional dimensions of quality (how it is supposed to *be* versus what it is supposed to *do*) -- for example, usability, scalability, performance, compatibility, reliability -- can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

## **TYPES OF TESTING**

### **Static vs. dynamic testing**

There are many approaches to software testing. Reviews, walkthroughs or inspections are considered as static testing, whereas actually executing programmed code

with a given set of test cases is referred to as dynamic testing. The former can be, (and unfortunately in practice often is...) omitted, whereas the latter takes place when programs begin to be used for the first time - which is normally considered the beginning of the testing stage. This may actually begin before the program is 100% complete in order to test particular sections of code (modules or discrete functions). For example, Spreadsheet programs are, by their very nature, tested to a large extent "on the fly" during the build process as the result of some calculation or text manipulation is shown interactively immediately after each formula is entered

## **1. UNIT TESTING:**

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

## **2. INTEGRATION TESTING:**

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

## **3. SYSTEM TESTING:**

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

### **System testing consists of the following steps:**

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

#### **4. FIELD TESTING:**

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned; we haven't tested our project using the field testing.

#### **5. ACCEPTANCE TESTING:**

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfils the goals or not.

#### **WHY SYSTEM TESTING?**

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. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

#### **ACTIVITY NETWORK FOR SYSTEM TESTING**

**The test plan entails the following activities:**

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile/assemble programs.
7. Prepare job performance aids.

8. Prepare operational documents.

**PREPARE TEST:** A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

- Outputs expected from the system.
- Criteria for evaluating outputs.
- A volume of test data.
- Procedure for using test data.
- Personnel and training requirements.

### **SPECIFY CONDITIONS FOR USER ACCEPTANCE TESTING**

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

### **PREPARE TEST DATA FOR PROGRAM TESTING**

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

### **PREPARE TEST DATA FOR TRANSACTION PATH TESTING**

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

### **PLAN USER TRAINING**

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

## **COMPILE / ASSEMBLE PROGRAMS**

All programs have to be compiled / assembled for testing.

## **PREPARE JOB PERFORMANCE AIDS**

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

## **PREPARE OPERATIONAL DOCUMENTS**

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

## **SYSTEMS TESTING**

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

## **ACCEPTANCE TESTING**

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

2. Senior Programmer
3. Junior Programmers
4. On line Users.

**Since this Project will complete in 4 months**

**COST ESTIMATE:** (Salary of Project Manager + Salary of Senior Programmer + 2 \* Salary of Junior Programmer) \* 2

## **9. GANTT & PERT CHART**

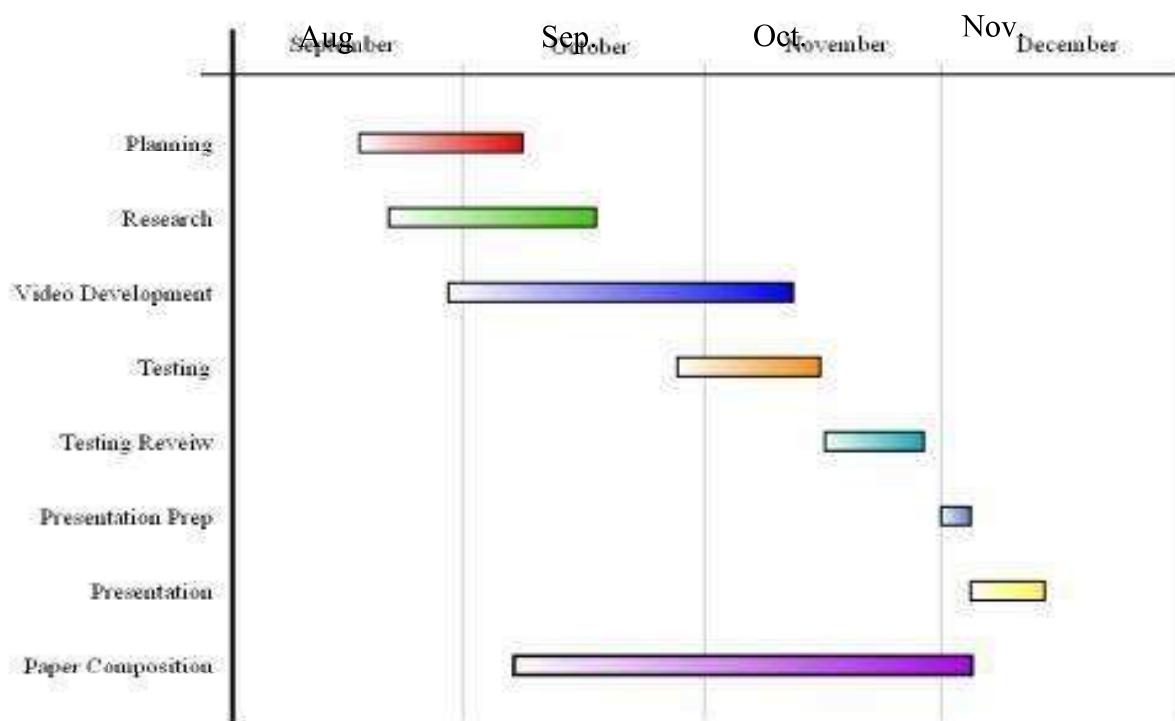
### **GANTT CHART**

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The

length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by colouring each milestone when completed. The project will start in the month of March and end after 4 months at the end of June.

## Gantt Chart



## Pert Chart

**PERT** (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.

**PERT** chart represents the statistical variations in the project estimates assuming a normal distribution. Thus, in a PERT chart instead of making a single estimate for each task, *pessimistic*, *likely*, and *optimistic* estimates are also made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. Since all possible completion times between the minimum and maximum durations for every task have to be considered, there are many critical paths, depending on the permutations of the estimates for each task. This makes critical path analysis in PERT charts very complex. A critical path in a PERT chart is shown by using thicker arrows. The PERT chart representation of THE PERFECT TRAINER MANAGEMENT SYSTEM problem of

Figure  
shown  
B.

A. is  
in Figure

Task	ES	EF	LS	LF	ST
<b>Specification Part</b>	0	15	0	15	0
<b>Design Database Part</b>	15	60	15	60	0
<b>Design GUI Part</b>	15	45	90	120	75
<b>Code Database Part</b>	60	165	60	165	0
<b>Code GUI Part</b>	45	90	120	165	75
<b>Integrate and Test</b>	165	285	165	285	0
<b>Write User Manual</b>	15	75	225	285	210

**Figure A:**

Different Tasks for the **THE PERFECT TRAINER**

**MANAGEMENT SYSTEM** are shown in above table.

## 10. SECURITY MEASURES

In this project we have used following validation checks.

## **8. COST ESTIMATION OF THE PROJECT**

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (PM).

### **Total No. Of Persons Involved In This Project:**

1. Administrator

2. Senior Programmer
3. Junior Programmers
4. On line Users.

**Since this Project will complete in 4 months**

**COST ESTIMATE:** (Salary of Project Manager + Salary of Senior Programmer + 2 \* Salary of Junior Programmer) \* 2

## **9. GANTT & PERT CHART**

### **GANTT CHART**

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The

## Pert Chart

**PERT** (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.

**PERT** chart represents the statistical variations in the project estimates assuming a normal distribution. Thus, in a PERT chart instead of making a single estimate for each task, *pessimistic*, *likely*, and *optimistic* estimates are also made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. Since all possible completion times between the minimum and maximum durations for every task have to be considered, there are many critical paths, depending on the permutations of the estimates for each task. This makes critical path analysis in PERT charts very complex. A critical path in a PERT chart is shown by using thicker arrows. The PERT chart representation of THE PERFECT TRAINER MANAGEMENT SYSTEM problem of

Figure  
shown  
B.

A. is  
in Figure

Task	ES	EF	LS	LF	ST
<b>Specification Part</b>	0	15	0	15	0
<b>Design Database Part</b>	15	60	15	60	0
<b>Design GUI Part</b>	15	45	90	120	75
<b>Code Database Part</b>	60	165	60	165	0
<b>Code GUI Part</b>	45	90	120	165	75
<b>Integrate and Test</b>	165	285	165	285	0
<b>Write User Manual</b>	15	75	225	285	210

**Figure A:**

Different Tasks for the **THE PERFECT TRAINER**

**MANAGEMENT SYSTEM** are shown in above table.

## 10. SECURITY MEASURES

In this project we have used following validation checks.

- While entering the data into the form it will check for the name of the client is properly filled & it should not be null.
- Whenever we enter the data for the new customer, company, or user will automatically check the details from the database tables and also generate the connection number automatically.
- Similarly, in the complaint table complaint number will generate automatically.
- Entered text / number should not exceed the limit (width).
- Almost for all fields we have used the validation for example if name of the fields requires the text type of data then it will check for the string and if the data is numeric then it will check if the number entered is proper numeric or not.

#### **JSP Provides Security**

- Evidence-based security (authentication)
- Based on user identity and code identity
- Configurable policies
- Imperative and declarative interfaces

## **11. SCOPE OF FUTURE APPLICATION**

\* This Project should be properly introduced and advertised to the user, so that its popularity increases over time.

\* This project should be fully Desktop based so that anyone can access using search engine.

\* Coordination between Member and Administration should be enhancing the usability of this project.

\* In future we provide this system accessible as a mobile app format so that user can use it very easily.

\* SMS and Email facility Provided to every tour operator, so that they get in touch with their clients.

\* Provide best services to the tourist and tour operator so that they happily use our system.

## **12. CONCLUSION**

This project is designed to meet the requirements of the enterprise. The software project is very big and includes almost all aspects of herbal store online application. It has been developed in JSP, keeping in mind the specifications of the system.

For designing the system, we have used simple data flow diagrams.

Overall the project teaches us the essential skills like:

- Using system analysis and design techniques like data flow diagram in designing the system.
- Understanding the database handling and query processing using My SQL.

## **13. BIBLIOGRAPHY**

### **Books:**

System Analysis and Design(IGNOU)

Software Engineering (IGNOU)

Java (Khalid Mughal)

### **Websites:** [www.javatpoint.com](http://www.javatpoint.com)

[www.jsptpoint.com](http://www.jsptpoint.com)

[www.onlineacution.com](http://www.onlineacution.com)

[www.indiaacution.com](http://www.indiaacution.com)

[WWW.Chatgpt.com](http://WWW.Chatgpt.com)