**TABLE OF CONTENTS**

TITLE OF THE PROJECT

INTRODUCTION OF THE PROJECT

OBJECTIVE OF THE PROJECT

PROJECT CATEGORY

SYSTEM ANALYSIS

* DATA FLOW DIAGRAM
* E-R DIAGRAM

MODULE DESRIPTION

DATA DITIONARY

PROCESS LOGIC

TESTING PROCESS

REPORT GENERATION

APPLICATION ARCHITECTURE

SYSTEM REQUIREMENT SPECIFICATION

* COMPTER HARDWARE SPECIFICATION
* SOFTWARE ENGINEERING PARADISM

PROJECT SCEDULING

* GANTT CHART

PROJECT LIMITATION

FUTURE SCOPE

**TITLE OF THE PROJECT**

***Institute Management***

***System***

***(IMS)***

**Project Introduction**

The **Institute Mangment System (IMS)** is the systematic and the additional method of the Institute management for the use of the organization. This is very helpful and highly useful. The system help to organization to store and manage all the record of institute. The word the **Institute Mangment System** has now taken over to the Personnel Management by the term mentions the management of the Institute people.

The principle of the **Institute Mangment System** has changed which they are treating as the individual targets and the necessities.

The IMS consists of the GUI application that has been developed by **SWING** for the front end and the **JAVA** for back end programming. The **Oracle database** is provided to the application.

**Project Objective**

The objectives of IMS are to maximize the return on investment from the organization’s human capital and minimize financial risk. It is the responsibility of the **Institute Mangment System** to conduct these activities in an effective, legal, fair, and consistent manner.

The **Institute Mangment System** has most of the Facilities eg.

* IMS help **Institute** to registration of the student,employee**.**
* IMS help **Institute** to find the record of the student and emploee.
* IMS help **Institute** to Update and Delete the record of the student and emploee.
* IMS help **Institute** to send Message or Notification to Student and Employee.
* IMS help **Institute** to Add new course ,update Existing Course ,Delete Existing Record and Find existing record.
* IMS help **Institute** to find the course at the time of Faculty registration.
* IMS help **Institute** to find the course and batch timing and Faculty for the course at the time of student registration.
* IMS help **Institute** to view the total Expenditure and Profit .
* IMS help **Institute** to Generate Payment Slip at the time when student are Giving payment.
* IMS help **Institute** to Generate Salary Slip to Employee.

And So on.

The IMS has developed an effective recruitment process. It has set certain standards that need to be followed while Registration of an Student & Registration of an employee.

**Project Category**

IMS is being made as Desktop-application using JAVA with Swing. so it will comes into OOPS Category. Object Oriented Programming System (OOPS) is the most popular and most dominant information management system available today. The OOPS provide following feature for a project:-

* **Inheritance (Codes Reusability):-** The concept of a data class makes it possible to define subclasses of data objects that share some or all of the main class OOP forces a more thorough data analysis, reduces development time, and ensures more accurate coding.
* **Abstraction (Information Hiding):-** Abstraction is simplifying complex reality by modeling classes appropriate to the problem, and working at the most appropriate level of inheritance for a given aspect.
* **Polymorphism (Overloading & overriding):-** It allows the programmer to treat derived class members just like their parent class's member.

This is a complete client server n tear architectural mechanism where Client sever describe the relationship among nodes in which one program the client makes a service request from another program, the server, which fulfills the request. Here client will be application or from end, which will generate request, and the server will be relational database management system which fulfills the request of clients by using client – server mechanism.

A Database system provides the enterprise with centralized control of its operation. Advantages for having centralized control of data are as follows:

* Redundancies elimination
* Inconsistency can be avoided
* The data can be shared
* Integrity can be maintained
* Conflicting requirements can be fulfilled

**System Analysis**

**Data Model:**

Data model is a tool that provides the ability to represent the data objects, their characteristics and their relationships to the software engineer/developer. It also provides an automated means for creating comprehensive Entity-relation Diagrams, Data Flow Diagram, Data object dictionaries and other related models. It also helps to understand the flow of data for user and easy to get any process from the system in any place.

**Entity Relationship Diagram:**

Entity Relationship Diagram is a pictorial representation in which the all attributes of an organization and their data fields are connected. There is a relation between two attributes; the relation is either one to many, many to one or many-to-many. There is also some special type of relations like generalization, specialization.

**Data Flow Diagram:**

The Data Flow Diagram (DFD) is also a pictorial representation which is divided into many parts like 0 Level, 1st level, 2nd level, 3rd level etc. A standard DFD is divided into at least three levels. It also shows the flow of data through a system. The system may be a company, an organization, a set of procedures, a computer hardware system, a software system or any combination of the proceeding. The DFD is also known as a data flow graph or a bubble sort. The following observations about DFDs are important:-

1. All names should be unique. This makes easier to refer to items of DFD
2. Defer error conditions and error handling until the end of the analysis.

**Data Object Dictionary:**

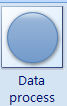
The Data Object Dictionary is a tabular representation of a database which is used in project. In this format the all data field of a data table described with primary key, data type, constraints, data size, null or not null etc. it is the tool for analysis, it stores data about data. It defines each data associated with the system.

**Data Flow Diagram**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. The Data flow Diagram shows the flow of data. It is generally made of symbols given below:-

* A square shows the Entity :



* A Circle shows the Process : 
* An open Ended Rectangle shows the data store : 
* An arrow shows the data flow :

**Module Description**

There are Ten types of modules:-

1. Student
2. Employee
3. Message
4. Library
5. Course
6. Inventory
7. Finance
8. Admin
9. Backup & Recovery Of IMS(institute management system) or software.

**[1] Student:-**

* This module is responsible to New Registration.
* This module is responsible to Delete Student Record.
* This module is responsible to Update Student Record.
* This module is responsible to Find Student Detail.
* This module is responsible to manage fees of student.

**[2] Employee :-**

* This module is responsible to New Registration.
* This module is responsible to Delete Employee Record.
* This module is responsible to Update Employee Record.
* This module is responsible to Find Employee Detail.
* This module is responsible to manage Salary of Employee.

**[3]. Message:-**

* This module is responsible to Send Email to Everyone.

**[4]. Library :-**

* This module is responsible to Add New Book.
* This module is responsible to Delete Book Record.
* This module is responsible to Update Book Record.
* This module is responsible to Find Book Detail.

**[5]. Courses :-**

* This module is responsible to Add New Course.
* This module is responsible to Delete Course Record.
* This module is responsible to Update Course Record.
* This module is responsible to Find Course Detail.

**[6]. Inventory :-**

* This module is responsible to Add New Goods.
* This module is responsible to Delete Goods Record.
* This module is responsible to Update Goods Record.
* This module is responsible to Find Goods Detail.

**[7]. Finance :-**

* This module is responsible to Generte Salary Slip.
* This module is responsible to Generate Fees Slip.
* This module is responsible to Manage Goods Expensense.
* This module is responsible to Manage Books Expensense.
* This module is responsible to Generate Expenditure and Profit Report.

**[8]. Admin:-**

* This module is responsible to registration of admin.
* This module is responsible to change Password.
* This module is responsible to update admin detail.
* This module is responsible to View admin detail.
* This module is responsible to logout from current session.
* This module is responsible to logoff from current session.

**[9]. Backup & Recovery Of IMS(institute management system) or software:-**

* This module is responsible to take backup and perform recovery of software.

**Data Dictionary**

1. Admin:-

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| NAME | NOT NULL | varchar2(40) |
| EMAIL ID | UNIQUE | varchar2(50) |
| PASSWORD | NOT NULL | varchar2(40) |
| DOB | NOT NULL | varchar2(11) |
| IMAGE\_PATH | NOT NULL , UNIQUE | varchar2(100) |

1. Batch\_Details:-

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| Employeeid | NOT NULL | varchar2(50) |
| Course\_id | Primary key | varchar2(50) |
| Batch\_timeing | NOT NULL | varchar2(20) |

1. **Books:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| purchasing\_date | NOT NULL | varchar2(11) |
| book\_name | NOT NULL | varchar2(40) |
| book\_id | Primary key | varchar2(50) |
| ISBN |  | varchar2(51) |
| Book\_category | NOT NULL | varchar2(30) |
| language | Not null | varchar2(20) |

1. **Course:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| Course\_duration | NOT NULL | int |
| course\_fees | Not null | int |
| course\_id | Primary key | varchar2(50) |
| course\_name | Not null | varchar2(40) |

1. **Employee:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| Salary | NOT NULL | INT |
| Dob | Not null | VARCHAR2(20) |
| JOB\_TIMEING | NOT NULL | varchar2(19) |
| NAME | NOT NULL | varchar2(40) |
| POSITION\_NAME | NOT NULL | VARCHAR2(40) |
| FATHER\_NAME |  | VARCHAR2(40) |
| MARITIAL\_STATUS | NOT NULL | VARCHAR2(10) |
| CONTACT\_NUMBER | UNIQUE AND NOT NULL | varchar2(15) |
| EMPLOYEEID | PRIMARY KEY | varchar2(50) |
| CATEGORY |  | VARCHAR2(10) |
| GENDER | NOT NULL | VARCHAR2(10) |
| RESIDENTAL\_ADDRESS |  | VARCHAR2(70) |
| ADDRESS |  | VARCHAR2(50) |
| ZIPCODE | NOT NULL | INT |
| STATE | NOT NULL | VARCHAR2(20) |
| CITY | NOT NULL | VARCHAR2(15) |
| COUNTRY | NOT NULL | VARCHAR2(30) |
| IMAGE\_PATH | UNIQUE NOT NULL | VARCHAR2(100) |
| EMAIL\_ID | UNIQUE | VARCHAR2(50) |

1. **Student:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| Sumbittedfess | NOT NULL | INT |
| Dob | Not null | VARCHAR2(11) |
| NAME | NOT NULL | varchar2(40) |
| FATHER\_NAME |  | VARCHAR2(40) |
| MARITIAL\_STATUS | NOT NULL | VARCHAR2(10) |
| CONTACT\_NUMBER | UNIQUE AND NOT NULL | varchar2(15) |
| Studentdid | PRIMARY KEY | varchar2(50) |
| CATEGORY |  | VARCHAR2(10) |
| GENDER | NOT NULL | VARCHAR2(10) |
| RESIDENTAL\_ADDRESS |  | VARCHAR2(70) |
| ADDRESS |  | VARCHAR2(50) |
| ZIPCODE | NOT NULL | INT |
| STATE | NOT NULL | VARCHAR2(20) |
| CITY | NOT NULL | VARCHAR2(15) |
| COUNTRY | NOT NULL | VARCHAR2(30) |
| IMAGE\_PATH | UNIQUE and NOT NULL | VARCHAR2(100) |
| EMAIL\_ID | UNIQUE | VARCHAR2(50) |
| faculty\_id | NOT NULL | Varchar2(50) |
| Coursed | Not null | Varchar2(50) |

1. **Goods:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| product\_name | NOT NULL | varchar2(40) |
| warranty\_duration |  | int |
| product\_desciption | NOT NULL | varchar2(100) |
| purchase\_Date | NOT NULL | VARCHAR2(15) |
| tin\_number |  | VARCHAR2(50) |
| goods\_id | Primary | varchar2(50) |
| image\_path | Unique not null | varchar2(100) |
| shopname | Not null | VARCHAR2(50) |
| warranty\_Type |  | VARCHAR2(20) |

1. **issue\_Books:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| book\_id | Primary key | varchar2(50) |
| issue\_date | Not null | Varchar2(11) |
| reciver\_id | NOT NULL | varchar2(50) |

1. **Qualification\_details:-**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Constraints** | **Data Type** |
| complete\_year | Not null | Int |
| Percentage |  | Int |
| institution\_name | NOT NULL | Varchar2(40) |
| stream |  | Varchar2(15) |
| Grade |  | Varchar2(2) |
| course\_name | Not null | Varchar2(20) |
| emp\_stu\_id | Not null | Varchar2(50) |

CREATE SEQUENCE Books\_id START WITH 1 INCREMENT by 1;

CREATE SEQUENCE employee\_id START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE course\_id START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE student\_id START WITH 1 INCREMENT BY 1;

CREATE SEQUENCE Goods\_Id START WITH 1 INCREMENT BY 1;

create table admin

(

name varchar2(40) not null,

email\_id varchar2(50) unique,

password varchar2(40) not null,

dob VARCHAR2(50) not null

);

create table books

(

purchasing\_date varchar2(20) not null,

book\_name varchar2(100) not null,

book\_id number primary KEY,

ISBN varchar2(50),

Book\_category varchar2(40) not null,

book\_language varchar2(30) not null

);

create table issue\_books

(

book\_id number primary KEY,

issue\_date varchar2(11) not null,

reciver\_id number not null,

FOREIGN KEY (book\_id) REFERENCES books(book\_id)

);

create table employee

(

salary int not null,

dob varchar(11) not null,

job\_timeing varchar2(19) not null,

name varchar2(40) not null,

position\_name varchar2(40) not null,

father\_name varchar2(40),

maritial\_status varchar2(10) not null,

contact\_number varchar2(15) not null,

id number primary key,

email\_id varchar2(50) unique not null,

category varchar2(10),

Gender varchar2(10) not null,

address varchar2(100),

zipcode int not null,

state varchar2(20) not null,

city varchar2(15) not null,

country varchar2(30) not null,

image\_path varchar2(100) unique not null

)

create table goods

(

product\_name varchar2(40) not null,

warranty\_duration int,

product\_desciption varchar2(100) not null,

purchase\_Date varchar2(50) not null,

tin\_number varchar2(50),

goods\_id number primary key,

shopname varchar2(50) not null,

warranty\_Type varchar2(40),

quantity number

);

create table student

(

batch\_timeing varchar2(70) not null,

totalFees int not null,

reaminFees int not null,

dob varchar(11) not null,

name varchar2(40) not null,

father\_name varchar2(40),

maritial\_status varchar2(10) not null,

contact\_number varchar2(15) not null,

id number primary key,

email\_id varchar2(50) ,

category varchar2(10),

Gender varchar2(10) not null,

address varchar2(100),

zipcode int not null,

state varchar2(20) not null,

city varchar2(15) not null,

country varchar2(30) not null,

image\_path varchar2(100) ,

teacherName varchar2(50) not null,

courseName varchar2(50) not null

);

create table course

(

Course\_duration int not null,

course\_fees int not null,

course\_id number primary key,

course\_name varchar2(40) not null

);

create table batch\_details

(

id number not null,

COURSE\_NAME varchar2(40) not null,

batch\_timeing varchar2(20) not null,

FOREIGN KEY (id) REFERENCES employee(id)

)

create table qualification\_details\_STUDENT(

Course varchar2(30),

complete\_year int not null,

percentage varchar2(10),

institution\_name varchar(40) not null,

stream varchar(15),

grade varchar(13),

course\_name varchar2(20),

id number not null,

FOREIGN KEY (id) REFERENCES STUDENT(id)

);

create table qualification\_details\_EMPLOYEE(

Course varchar2(30),

complete\_year int not null,

percentage varchar2(10),

institution\_name varchar(40) not null,

stream varchar(15),

grade varchar(13),

course\_name varchar2(20),

id number not null ,

FOREIGN KEY (id) REFERENCES employee(id)

);

**Process Logic**

* In this Desktop Application Admin can login and perform many tasks like view his/her profile, New Regisration of student and employee, Send message to employee and student,Manage Employee and student record.and etc.
* For this process Admin Need to Register/signup first time. And then he/she can login and can perform many task.

**Testing Process**

Testing is a process used to measure the quality of developed software. Software testing is a process or a series of processes designed to make sure code what it was designed to do. Software should be predictable and consistent, offering no surprise to users. At the start of testing a program shouldn’t be merely tested to show that it works. Rather the test should start with the assumption that the program contains errors and the goal is to find as many errors as possible. Hence more appropriate definitions are:

* Testing is a process to executing a program with the intent of finding errors.
* Testing is the process uncovering errors in a program make it a feasible.
* Testing is a destructive process of trying to find the errors in a program.
* It is used to determine the quality of product according to the users need.
* Testing is a process to demonstrating that errors are not present.
* The purpose of testing to show that a program performs its intended functions correctly and a process of establishing confidence that a program does what is it supposed to do.

Once source code has been generated, software must be tested to uncover the error before utilization of software or deliver. **Software is tested on the following strategies**: It is impractical and

often impossible to find all the errors in a program. This fundamental problem in testing thus throws open the question as to what would be the strategy that you would adapt to test. Two of the most prevalent strategies include **Black box testing** and **white box testing / Glass box testing.** Internal program logic is exercised using “White box / Glass box” test case design technique. In this testing also examine that the fundamental aspect of the system with complete information and access to the internal logical structure, code and algorithms. Using White box / Glass box test methods we derive test cases that:

* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false side.

Software requirements are exercised using “Black box” test case design techniques.

Black box tests are designed to validate functional requirements without regard to the internal working of a program. Black box testing is not an alternative to white box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white box methods. Back box testing attempts to find errors in the following categories:

* Incorrect or missing functions and Interface error.
* Errors in data structure or external data base access.
* Behavior or performance errors, and Initialization and terminate errors.

Testing is a set of activities that is planned in advance and conducted systematically.

1. **Validation and verification**

Verification refers to the set of activities that insure that s/w correctly implements a specific function. Validation refers to Different set of activities that ensure that the s/w that has been built is traceable to customer requirements.

1. **Unit testing**

It focuses on verification of s/w components or module. It is a white box oriented and steps are conducted in parallel for multiple components.

1. **Integration testing**

After unit testing each module is integrated as whole and at the point. Some problems may arise in interfacing the strategy of integration may be of two types: Non-incremental approach and Incremental approach.

* 1. **Non-incremental approach:** All components are combined and their entire program is tested.
  2. **Incremental approach:** Program is constructed and tested in small increment so it’s easier to isolate error.

1. **Validation testing:**

Validation testing is the final stage of s/w tests. Validation can be defined in various ways but the simple definition is that validation succeed when s/w function in a manner that can be reasonably e7ected by the customer.

**Report Generation**

Following are the reports that will generated in this IMS Desktop application:-

* Employee Salary Report :-
* Admin can view the details of salary status that are paid by institute.
* Employee Details Report :-
* Admin can view the details of employee.
* Student Fees Report :-
* Admin can see the detail of the student.
* Student Details Report :-
* Admin can see the detail of student.
* Inventory Details Report :-
* Admin can see the details of goods.
* Course Details Report :-
* Admin can see the details of Course.
* Library Details Report :-
* Admin can see the details of Books.
* Finance Report :-
* Admin can see the details of Finace (Expenditure and profit).
* Admin Details Report :-
* Admin can see the His/her self details of Course.

**APPLICATION ARCHITECTURE**

The project “**INSTITUTE MANAGMENT SYSTEM**” is developed using N-Tier Architecture.

*N-tier* data applications are data applications that are separated into multiple *tiers*. Also called "distributed applications" and "multitier applications," n-tier applications separate processing into discrete tiers that are distributed between the client and the server. When you develop applications that access data, you should have a clear separation between the various tiers that make up the application.

A typical n-tier application includes a presentation tier, a middle tier, and a data tier.

The easiest way to separate the various tiers in an n-tier application is to create discrete projects for each tier that you want to include in your application. For example, the presentation tier might be a Windows Forms application, whereas the data access logic might be a class library located in the middle tier. Additionally, the presentation layer might communicate with the data access logic in the middle tier through a service such as a service. Separating applica tion components into separate tiers increases the maintainability and scalability of the application. It does this by enabling easier adoption of new technologies that can be applied to a single tier without the requirement to redesign the whole solution.

In addition, n-tier applications typically store sensitive information in the middle-tier, which maintains isolation from the presentation tier.

## C:\Users\pro\Desktop\n-tier-architecture-1.pngPresentation Tier

The presentation tier is the tier in which users interact with an application. It often contains additional application logic also.

The presentation tier does not directly access the data tier. The presentation tier communicates with the data tier by way of the data access component in the middle tier.

## Middle Tier

The middle tier is the layer that the presentation tier and the data tier use to communicate with each other. Typical middle tier components include the following:

* Business logic, such as business rules and data validation.
* Data access components and logic, such as the following:
  + Common application services, such as authentication, authorization, and personalization.

The following illustration shows features and technologies that are available in Visual Studio and where they might fit in to the middle tier of an n-tier application.

The middle tier typically connects to the data tier by using a data connection. This data connection is typically stored in the data access component.

## Data Tier

The data tier is basically the server that stores an application's data (for example, a server running Oracle Server).

The following illustration shows features and technologies that are available in Visual Studio and where they might fit in to the data tier of an n-tier application.

**System Requirement Specification**

A project is developed with the help of tools and platform. Tools and platform is very necessary component for Developer, without tools and platform it is impossible to develop any project.

We have used to java(back-end) and Swing(front-end) and oracle 11g (database) for this project.

**Front-end (Swing)**

* Swing is a part of Java Foundation Classes (JFC).that is used to create Desktop-based applications.
* It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.
* Unlike AWT, Java Swing provides platform-independent and lightweight components.
* The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.
* The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.
* Swing was developed to provide a more sophisticated set of GUI [components](https://en.wikipedia.org/wiki/Software_component) than the earlier [Abstract Window Toolkit (AWT)](https://en.wikipedia.org/wiki/Abstract_Window_Toolkit).
* Swing provides a native [look and feel](https://en.wikipedia.org/wiki/Look_and_feel) that emulates the look and feel of several platforms, and also supports a [pluggable look and feel](https://en.wikipedia.org/wiki/Pluggable_look_and_feel) that allows applications to have a look and feel unrelated to the underlying platform.
* It has more powerful and flexible components than AWT.
* In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.
* Unlike AWT components, Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such an element

**Back-end (Java):-**

* Java is a general-purpose [computer programming language](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), and specifically designed to have as few implementation dependencies as possible.
* JAVA help application developers "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere)" (WORA). meaning that [compiled](https://en.wikipedia.org/wiki/Compiler) Java code can run on all platforms that support Java without the need for recompilation.
* Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture).
* As of 2016, Java is one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) particularly for client-server web applications, with a reported 9 million developers.
* The language derives much of its [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.
* The latest version is [Java 8](https://en.wikipedia.org/wiki/Java_version_history) Update 121 which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.
* As of May 2007, in compliance with the specifications of the [Java Community Process](https://en.wikipedia.org/wiki/Java_Community_Process), Sun [relicensed](https://en.wikipedia.org/wiki/Software_relicensing) most of its Java technologies under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License).
* Java is very secure programming language.

**ORACLE 11G(DATABASE):-**

* Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an [object-relational database management system](https://en.wikipedia.org/wiki/Object-relational_database_management_system) produced and marketed by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation).
* An Oracle database system—identified by an alphanumeric system identifier or SID—comprises at least one [instance](https://en.wikipedia.org/wiki/Database) of the application, along with data storage.
* An instance—identified persistently by an instantiation number (or activation id: SYS.V\_$DATABASE.ACTIVATION#)—comprises a set of operating-system [processes](https://en.wikipedia.org/wiki/Process_(computing)) and [memory](https://en.wikipedia.org/wiki/Computer_data_storage)-structures that interact with the [storage](https://en.wikipedia.org/wiki/Computer_data_storage).
* Users of Oracle databases refer to the server-side memory-structure as the SGA (System Global Area).
* The SGA typically holds [cache](https://en.wikipedia.org/wiki/Cache_(computing)) information such as data-buffers, [SQL](https://en.wikipedia.org/wiki/SQL) commands, and user information.
* The [Oracle RAC](https://en.wikipedia.org/wiki/Oracle_RAC) (Real Application Clusters) option uses multiple instances attached to a central [storage array](https://en.wikipedia.org/wiki/Disk_array).
* The Oracle RDBMS [stores data logically](https://en.wikipedia.org/wiki/Data_storage_device) in the form of [tablespaces](https://en.wikipedia.org/wiki/Tablespace) and physically in the form of data [files](https://en.wikipedia.org/wiki/Computer_file) ("[datafiles](https://en.wikipedia.org/wiki/Datafile)").

**Computer Hardware And Software Specification**

According to available Hardware Components in the market their some essential hardware components for this Desktop Offline application “**Institute Management System**” which are as follows:

* Microprocessor: Intel Dual Core (2.66 GHZ)
* Operating System: Anyone(Linux, Mac,Windows)
* Hard-Disk: 50 GB(Size need to be increase According to Data)
* RAM: 2 GB
* Monitor: Color
* Internet Connection(Only for Sending message).

**Software Engineering Paradigm:**

To give a standard and efficient mechanism in the software development an appropriate paradigm must be implemented. In the development of my project the “**Incremental Model**” is adopted, because the Incremental Model applies Liner Sequence fashion as a calendar time progresses. When an incremental model is used in the first increment the basic requirements (core product) are addressed, but many supplementary features (unknown) remain undelivered.

**PROJECT SCHEDULING**

**Analysis**

**Design**

**Test**

**Code**

**Analysis**

**Design**

**Test**

**Code**

**Analysis**

**Design**

**Test**

**Code**

**Analysis**

**Design**

**Test**

**Code**

**Increment 1**

**Increment 2**

**Increment 3**

**Increment 4**

**Delivery of 1st**

**Increment**

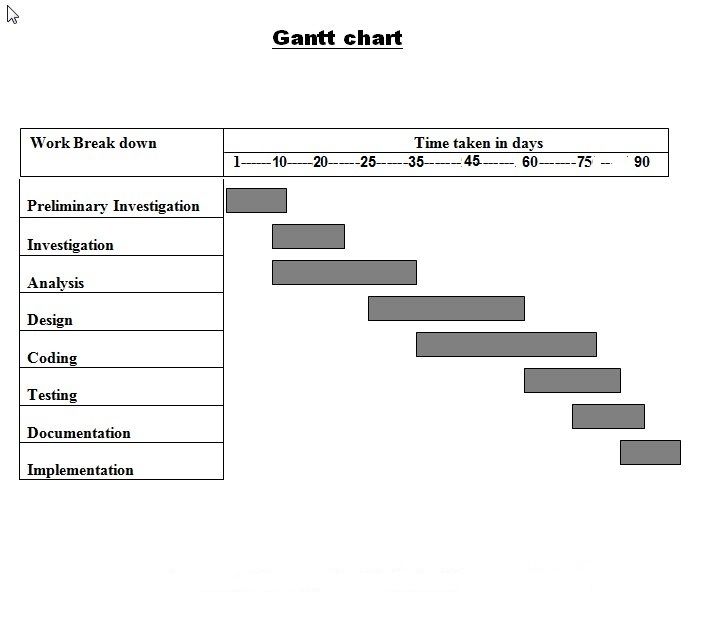
**Delivery of 2nd**

**Increment**

**Delivery of 3rd**

**Increment**

**Delivery of 4th increment**

****

**Gantt Chart for “Institute Managment Management System”**

**Project Limitations**

Every System has a limitation at a fixed point. Because a single system can’t satisfies all the requirements of user for a long time of period. Any system is developed according to the view of both sides (Developer & User) requirements. Before developing of system all factors/limitation are given by vendor. The following limitations of the project “**Institute Management System**” is:

* The software developed for the stated objectives, contains the essential features but failed to generate the entire user requirements.
* When measured the quality matrix the software lacks the necessary flexibility sure when weighted average is calculated.
* The prototype generated for the purpose lacks the necessary structuring due to ine7erience.

These all limitations can be corrected either by modification in the appropriate modules, entity or to give a proper training to their users. And these all objectives are helpful for user to working in proper manner and without any difficulties.

**Future Scope**

The scope is the most important element to understand about any project. All planning and allocation of resources are anchored to this understanding. The future scope of a software is completely depends on the fulfillment of requirements and management of database in sense of not only storing & accessing but also searching, sorting and updating the database. I will develop my Offline software using oracle 11g database, which is best for a medium level and high level Organization. Oracle develops both tools front-end and back-end, so there is no confliction between database connectivity. So, the database is properly managed & maintained. All types of reports and statements will easily generate and lots of data are handled in a systematic manner. I will try to implement back up, data recovery, and zipping mechanism also in my proposed system.

* This GUI Application can be further modify to a large scale organization.
* Each module of the application can be further modified & extended. Such as recruitment, planning etc.
* The application can be extended by enhancing Training and development process, performance analysis and management, promotion criteria etc.