

**A**

**PROJECT**

**ON**

# 

PLACEMENT MANAGEMENT

SYSTEM



SUBMITTED BY:-

NYAYAPATI MANASI

UNDER THE GUIDANCE OF:-

BETA CENTUARI

**ABSTRACT**

The project named ―”Placement Management System”, a student/company information system is a web based system. The project is developed on the basis of ―Placement Cell being presently used in the University for storing and retrieving the information of students and companies who are registered in Placement Cell. The Placement Cell maintains a large database of students wherein all the information of student including the personal records and the academic performance in terms of the SPI and PPI is stored and company information including profile of company, eligibility criteria and facilities it provide etc. The software retrieves this data and displays as per the user requirement.

The Placement Management System is developed as an attempt to take a record of company and students by restricting such a large database to that of a particular class of students or company. The System provides the facility of viewing both the personal and academic information of the student and company. It can also search for eligible students and company and also insertion and deletion of records by the administrator. Eligible students will receive an email including the details of the company and also they will receive a sms on their registered mobile number in which the date-time and location will be provided.

**Table of Content**

1. Introduction

1.1 Objective (Functionalities of the system)

1.2 Scope

1.3 About the present system

1.4 Definition

1.5 Objective of the Project

1. System Analysis

2.1 Proposed System

2.2 System Design

2.3 Feasibility Study

2.4 Project Planning

2.5 Project Scheduling

2.5 System Requirements & Specifications

2.6 Functional Requirements

2.7 Software Engineering Paradigm Applied

2.8 DFD

2.9 Diagrams and Designs

3. System Design

3.1 E-R Diagram

3.2 Network Architecture

6. Implementations and Testing

7. Future Scope & Cost Estimation

8. Bibliography

Introduction

INTRODUCTION

**1.INTRODUCTION TO PLACEMENT MANAGEMENT SYSTEM**

**1.1.OBJECTIVE**

**Placement Management System** aims at providing the compatibility to simplify the process of placement for students. This system that consists of a student login, company login and an admin login. This is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and upload all their details including their marks onto the system. The admin can check each student details and can remove faulty accounts. The system also consists of a company login where various companies visiting the college can view a list of students in that college and also their respective resumes. The software system allows students to view a list of companies who have posted for vacancy. The admin has overall rights over the system and can moderate and delete any details not pertaining to college placement rules.

**1.2.SCOPE**

This system has a big scope to do. Students can maintain their information. Notifications are sent to students email address about the companies. Students can access previous information about recruitment. This project has a large scope as it has the following features which help in making it easy to use, understand and modify it:

• Automation of Placement Procedure

• No Need to do Paper Work

• To save the environment by using paper free work

• To increase the accuracy and efficiency of placement procedure

• Management of Student Data

• Analysis of overall Placement

## 1.3.ABOUT PRESENT SYSTEM

## In Various colleges, training and placement officers have to manage the students profile and documents of students for their training and placement manually. Placement Officer have to collect the information of various companies who want to recruit students and notify students time to time about them. Placement Officer have to arrange profiles of students according to various streams and notify them according to company requirements. If any modifications or updates are required in the profile of any student, it has to searched and to be done it manually.

## 1.4.Definition

Placement Management System plays an indispensable part in reducing a recruitment agency’s administrative burden.

This project is to facilitate students in college, company to register and communicate with Placement Office. The users can access easily to this and the data can be retrieved easily in no time. In the main page there are options for a new register, a registered student to directly login using username and password, submit resume. In the student registration form, we can give personal details, educational qualifications, and professional skills and upload resume. The job details of the placed students will be provided by the administrator. The administrator plays an important role in our project. They provide approval of student registration and updating. In this project we create a search engine for administrator, who can search everything about the student and company.

**1.5.OBJECTIVES OF THE PROJECT**

Computers and information technology has a major influence on the society and the society is becoming more and more dependent on technology. Going on is an era of simplifying almost all complicated works using computers. The last few years have witnessed a tremendous increase in the capabilities and use of computers. Manual processing makes the process slow and other problems such as inconsistency and ambiguity on operations. The proposed system intends user-friendly operations which may resolve ambiguity. By considering all this factors, the applications produced, which performs the social service simply and effectively.

System Analysis

**PROPOSED SYSTEM**

This project is aimed at developing an online application for the placement department of the college. Once you open this web application at the front end all the schedule/event are available to everyone. The administrator is the master user; He gets the most number of priorities than the other users. The different functions involve the case of an administrator are updating, approval. The administrator can view and approve the various application forms. Students can edit their resume online, and update them constantly. When student fill the information for resume building they doesn’t need or allowed to enter their yearly engineering marks, it will be extracted from result analysis documents of college. Students can very flexibly search for and view company and vacancy details, and apply to vacancies by attaching a resume. Students can access relevant resources placed online for them. They will have access of forum, with which they can post queries, reply queries. The Placement Officer can give an approval to edited information done by the student. In addition, Placement Officer can search the eligible student based the company criteria and can generate the list. Also system can send an automated email to the eligible student. Placement Officer can communicate with the student through the forum module. When Placement Officer first login, they see all the recently changed / created companies and vacancies. Placement Officer can gather information on all students for which they have appropriate permissions. They can see the information provided by the student like personal details, educational details, parent mobile number, extracurricular activity or other information. They put online notices, schedule and events so that the entire user can view this. They can communicate with the Student or Placement Officer through forum. The companies will have to get register for the first time so that their information’s like the URL, the contact information, papers, vacancies will be provide. Companies can edit their own contact information to help keep it current. Companies can view all students that have applied for vacancies, together with information on their availability, application time, cover letters, attached resume etc. The proposed Online Placement system is intended to avoid all the drawbacks of existing system. It will add some more features than the existing system. The proposed Online Placement system is a cost effective way of doing the manual processes done in the existing system. This helps the organization to win the war in the existing competitive world.

A. Resume Building: Standardized format will be provided and student can use it to generate their resume. All students information will be available in the database .This information can be extracted automatically from the database for generating standard resume. This simplifies and speeding up the process of resumes building.

B. Mail Notification: As per company criteria E-mail message will be send to eligible student. So, that interested student can register for company directly by clicking on apply

button. Mail from system will be sent to each registered user keeping them updated all the time.

C . Report Generation: Analysis of Placed students, Un-placed students, department wise placement will be made available through this tool. Various algorithms can be used for placement analysis.

**SYSTEM DESIGN**

**An Overview to System Design**

Designing is the most important part of Software Development. It requires careful Planning and logical thinking on part of the system designer. Designing the software means to plan how the various parts of the software are going to meet the desired goals. It should be done with utmost care because if the phase contains any error then it will affect the performance of the system in the future. The system may take extra processing time, more coding and extra workload.

System design involves first logical design and physical design construction of the system. When an analyst formulates a logical design and writes the detail specifications for the new system, they describe the features, input, output, files, database and procedures. The statements of these features are termed as design specification, physical construction, the activities following logical, procedure program software, and file and work system. The design specifications instruct the programmers about what the system do. These programmers in turn write the program, accept the data from users, process data, produce report and store data.

**Phases of System Design**

Design of Output

Design of Input

Normalization Rules Followed

Design of Database

Design of Data Dictionary

**Design of Output**

Output is the most important and direct source of information to the user. Efficient, intelligent output design improves the relationship of the system with the user & helps in decision-making.

The output of the system consists of both printed as well as displayed output on the screen. The displayed output consist is mostly used during enquiry and registration process to show various details in response to queries of clients. Outputs are either printed or displayed mostly for the following purposes.

Information about assets

Information to user on submission of the complaint

Information to user on status of the order

**Design of Input**

Incorrect input data are the most common source of errors in data processing. Error induced by data entry operator can be controlled by intelligent input design. Input design is the process of converting user-oriented inputs to a computer-based format. Input to the system contain previously stored from the database as well as user specific entitles. Input generated through forms is stored in appropriate tables for processing. In the designed system most of the data comes from database. So entry errors are minimized at level of users. In all the four processes, which lead to one another, some data are entered always.

**Normalization**

Normalization is the process of organizing the database in such a manner that the relation between the entities becomes easier and redundancy is eliminated. Generally they are three normal forms, which are explicitly used in database design. They are 1NF, 2NF, and 3NF. Though normal forms having higher orders are available, but these are not used because they are used only in multi valued dependency cases. First three normal forms are used extensively because of their simplicity.

**FIRST NORMAL FORM**

A database is said to be in First Normal Form if it has not any repeating groups. A repeating group is one, which contains more than one set of values for some specific groups of values. Relational database does not allow repeating groups.

**SECOND NORMAL FORM**

A database is said to be Second Normal Form if it is in the First Normal Form and every nonkey attribute is fully dependent on primary key.

**THIRD NORMAL FORM**

A table or database is said to be in Third Normal Form if and only if it is Second Normal Form and all non-key field is non transitively dependent on Primary key i.e. there should not exit any transitive dependency among various non-key.

The Higher the degree of Normalization, The less redundant data your database will contain.

**Database Design**

The database is a mechanized, formally defined, centrally collected repository of data in an organization. The database design actually involves two steps. They are logical database design and physical database design. Logical database design deals with organizing the database in term of relationship among the entities. The physical database takes care of the actual organization of data and the access method.

* **LOGICAL DATABASE DESIGN**

The databases were recognized first from the information gathered about the user’s requirements the reports to be generated and interacting systems. The organization of fields strictly depends on the relationship. In an organizing the database, the standard naming convention as follow to improve the logical meaning and to make the task understandable to the user. Each field is named in such a way that it can be identified by its name. During the logical design the main consideration are:

* Data integrity**:**

The data is organized and decomposed in such a way that integrity of the data is maintained. Since the system interacts with the other systems, universal naming is followed within the system, which is an agreement with the interactive system.

* DataConsistency**:**

The database has been designed in such a way that it will maintain consistency and data integrity in a dynamic fashion. If any change is made to certain field of a table, then the value of all the instances of that field in other tables will also be updated to new.

* Data Redundancy**:**

The database has been designed to reduce the redundancy so that control over data is not lost and consistency is maintained through out the database.

* **PHYSICAL DATABASE DESIGN:**

The physical database design deals with how data is stored and how it is retrieved. Indexes help SQL server to locate data. They speed and how it is retrieval by pointing SQL server to where a table column data is stored on Stock. Table may have more than one index; SQL server supports the following indexes:

* Clustered and Non-clustered Indexes

Clustered force SQL server to continually sort and re-sort the row of the table so that their physical order is always the same as the logical order. You can have only one clustered index per table. Each non-clustered index can provide access to the data in a different sort order.

* Unique Indexes

These indexes do not permit any two rows in the specified column to have the same values server checks for duplicate values when the index is created and each time the data is added.

* Data Dictionary:

Data dictionary is a repository of data or in other word it is the data about data. The data dictionary holds information about all fields that have been used in the system. At any time the user can look at it for ready references.

* Logical design**:**

The logical design of the “Unified Material Code Management System” addresses the details that how a system will meet the requirement identified during system analysis. The issues focused are:

* Data flow diagram

The logical system model is termed as the Data Flow Diagram that addresses “what occurs in the system”. It shows the flow of data in the system between processes and data stores. The various graphical symbols used are:

**Database architecture**

Database Architecture of Online Recruitment System entitled “E-recruitment: As Choice Is Clear” can be best explained by using ER diagrams and Relational Database Diagram. The ER Diagrams and relational tables are designed for the development of this project. Finally table constraints and functional dependencies are also given at the last of this section.

**Data Integrity and Constraints**

Data integrity guarantees that the data in a database must follow to a pre-defined set of constraints. Data Integrity rules followed in our project for designing the database are:

* **Entity Integrity**

To maintain integrity of data within an entity

* **Referential Integrity**

To maintain integrity of data between tables

* **Column Integrity**

Maintaining values consistent with the defined data format

A good database design must also follow certain constraints to provide data consistency and reliability. The constraints used in our database design are:

* **Primary Key**

Ensures each row of a table must contain an entry that uniquely distinguishes the row from any other row.

* **Unique Key**

Ensures no two values in a column of a table may be the same.

* **Foreign Key**

Ensures that a row in one table may have a relationship to data in another table, link, or pointer, between related rows in the two tables is called the foreign key

* **Not Null**

Requires that values are provided whenever data is inserted for a specified column

* **Check**

Requires specific values for a column, or conditions for the values in a column

**System Design**

The system consists of three modules as admin module, student module and recruiter module. Each module has an same login page that contain user id and password field, by entering value in that field the user should login to the system. Fig. Block Diagram

1) Current Student: In the current student module, once the student login, they will see the news and events at the homepage. They can manage the profile by putting all the information. They can communicate with the other users. They can download the resources provided by the users. They can see the company registered, company website, criteria and other information provided by company.

2) Placement Officer: Placement Officer can prepare schedule & events about companies & flash it to the every student login as well as front page. They can contact with the company & can ask the detail information about company. They can take feedback & can see filled feedback given by alumni or company. Placement Officer can communicate with the users using forum.

3) Company: Company can register with the college. In company module contact information, URL, Papers, vacancies will be provided. Company can edit their own contact information. Company can see how many students are eligible based on the criteria provided. Company can take feedback from Placement Officer.

4) Department Staff: One authorized staff can register with the system. Staff can upload the information. They can participate in communication in communication through forum. Staff can see the any notice sent by Placement officer. Staff can see the schedule or event in their login. Staff can see the company & jobs created by admin. Staff can see the registered student & their status.

**FEASIBILITY STUDY**

All projects are feasible given unlimited resources and infinite time. Unfortunately the development of computer-based system in many cases is more likely to be plagued by scarcity of resources and delivery date. Hence, we have made use the concept of reusability that is what Object Oriented Programming (OOPS) is all about.

The feasibility report of the project holds the advantages and flexibility of the project. This is divided into three sections:

* Economic Feasibility
* Technical Feasibility
* Behavioural Feasibility

## Economic Feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to be determining the benefits and savings that are expected from a candidate and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

A systems financial benefit must exceed the cost of developing that system. I.e. a new system being developed should be a good investment for the organization. Economic feasibility considers the following

* The cost to conduct a full system investigation.
* The cost of hardware and software for the class of application.
* The benefits in the form of reduced cost or fewer costly errors.
* The cost if nothing changes (i.e. the proposed system is not developed).

The proposed “**PLACEMENT MANAGEMENT SYSTEM**” is economically feasible because

* The system requires very less time factors.
* The system will provide fast and efficient automated environment instead
* of slow and error prone manual system, thus reducing both time and man
* Power spent in running the system.
* The system will have GUI interface and very less user-training is required
* to learn it.
* The system will provide service to view various information for proper managerial decision making.

## Technical Feasibility:

Technical feasibility centers around the existing computer system (Hardware and Software etc) and to what extend it support the proposed addition. For example, if the current computer is operating at 80 percent capacity - an arbitrary ceiling - then running another application could overload the system or require additional Hardware. This involves financial considerations to accommodate technical enhancements. If the budgets are a serious constraint, then the project is judged not feasible. In this project, all the necessary cautions have been taken care to make it technically feasible. Using a key the display of text/object is very fast. Also, the tools, operating system and programming language used in this localization process is compatible with the existing one.

## Behavioral Feasibility:

People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. Therefore it is understandable that the introduction of a candidate system requires special efforts to educate and train the staff. The software that is being developed is user friendly and easy to learn. In this way, the developed software is truly efficient and can work on any circumstances, tradition, locales.

Behavioral study strives on ensuring that the equilibrium of the organization and status quo in the organization neither are nor disturbed and changes are readily accepted by the users.

**Request Approval**

The system was found to be technically, economically and operationally feasible. The system also has the support of both the user and the management. Hence the project was approved.

**PROJECT PLANNING**

Project planning is a discipline for stating how to complete a project within a certain timeframe, usually with defined stages, and with designated resources. One view of project planning divides the activity into:

* Setting objectives (these should be measurable)
* Identifying deliverables
* Planning the schedule
* Making supporting plans

Supporting plans may include those related to: human resources, communication methods, and risk management.

Computer hardware and software project planning within an enterprise is often done using a project planning guide that describes the process that the enterprise feels has been successful in the past.

**Developing a written plan** (and setting aside the necessary time for this) is essential to start involving partners and stakeholders in what you aim to do and to obtain funding. A clear plan will:

* provide relevant background to explain what the need or issue is that you want to address
* set out anticipated outcomes or changes that you expect to achieve
* map these into national and local policy contexts
* Set out in a format that is easy to read at a glance:
* your specific objectives
* actions you propose to take or services you will provide to meet these
* how they will be delivered
* by when
* by whom
* How you will know whether you have been successful or not.
* Involvement in the planning process

Identifying and making contact with key stakeholders is an important part of preparing your plan. Making contact at planning stage:

* will help you find out what is already going on in your area
* ensure your plan adds value to work in progress rather than duplicating existing projects
* open the door for participation in the development of your work from your target audience and from key stakeholders
* Help ensure your plan is realistic and achievable.

This approach is not only relevant to working with refugees and asylum seekers. It can be applied to planning any project or programme.

**PROJECT SCHEDULING**

Scheduling is an inexact process in that it tries to predict the future. While it is not possible to know with certainty how long a project will take, there are techniques that can increase your likelihood of being close. If you are close in your planning and estimating, you can manage the project to achieve the schedule by accelerating some efforts or modifying approaches to meet required deadlines.

One key ingredient in the scheduling process is experience in the project area; another is experience with scheduling in general. In every industry area there will be a body of knowledge that associates the accomplishment of known work efforts with time duration. In some industries, there are books recording industry standards for use by cost and schedule estimators. Interviewing those who have had experience with similar projects is the best way to determine how long things will really take.

When preparing a schedule estimate, consider that transition between activities often takes time. Organizations or resources outside your direct control may not share your sense of schedule urgency, and their work may take longer to complete. Beware of all external dependency relationships. Uncertain resources of talent, equipment, or data will likely result in extending the project schedule.

Experience teaches that things usually take longer than we think they will, and that giving away schedule margin in the planning phase is a sure way to ensure a highly stressed project effort. People tend to be optimistic in estimating schedules and, on average, estimate only 80% of the time actually required.

Failure to meet schedule goals is most often due to unrealistic deadlines, passive project execution, unforeseen problems, or things overlooked in the plan.

Most of the popular project management software packages enable the project resource planner to assign staff to project tasks, display resource requirements profiles, and adjust the schedule of slack tasks so resource requirements more closely fit those available in the organization. Some packages can display multiple project resource requirements to facilitate organization-wide resource management, optimization, and leveling. Individual project requirements may be adjusted by manipulating schedule slack in tasks not on the critical path. This can facilitate allocation and leveling of staff throughout the organization.

Efforts to accelerate a project schedule are commonly grouped under the term "crashing" the schedule. Maybe this term was coined to suggest that there is always some price for driving a project to completion sooner than normal. There are a number of ways to improve the schedule when your boss says, I need it sooner!

1. Add people to the schedule. Additional staff must be added early in a project or they will slow it down while learning the ropes. If you add people, you may also need to add staff for supervision and coordination, so staff is fully applied.

2. Improve productivity and work longer hours. A good team atmosphere with management support can help make this happen. Without positive nourishment of this process, you could lose your team to attrition.

3. Review schedule dependencies and look for opportunities to overlap tasks or make serial tasks concurrent or parallel activities. This requires greater coordination and sometimes involves increased risks which need to be managed carefully.

4. Review the project scope and remove or delay features or functionality from the project critical path.

5. Consider innovative approaches such as a different development methodology, alternative technologies, or out-sourcing options.

**SYSTEM REQUIREMENTS AND SPECFIFICATIONS**

**Determination of System Requirements**

Requirement determination involves studying the current business system to find out how it works and where improvements should be made. A requirement analysis is a feature that must be included in a new system. This includes a way to capture or process data, producing information, controlling a business activity or supporting management.

**Activities in requirement determination**

Requirement determination involves three major activities. These activities are:

* Requirement anticipation
* Requirement investigation
* Requirement specification

**Requirement Investigation*:***

This activity is the heart of system analysis. Requirement investigation relies on the fact-finding techniques. Fact-finding techniques are the specific methods that are used for collecting data about requirements. These include the interviews, questionnaire, record inspections and observations.

**Interview:**

Interviews were used to collect information from individuals and from the employees of the various unit individuals across the units. The people interviewed included the users of the existing system and the potential unstructured users of the proposed system. Interview can be structured or unstructured. The method of interview was used to acquire general information about the system.

**Record Review:**

The records and reports that are maintained and generated by the users of various personal of various units for E-Recruitment HR Solutions were viewed.

**Observations:**

Observation was used to obtain firsthand information about how activities are carried out. The way the documents were handled and processed were carried out were also observed.

To implement E-Recruitment HR Solutions, the basic things that are needed are:

1. **SOFTWARE REQUIREMENT:**

* Operating system :- Windows XP SP-3/ Window 7
* Database :-
* Connector:-
* Web Browser:- Internet Explorer 6.0/7.0
* Web Page Development :-
* Web Page Style sheet :- Html, Java Script, Ajax, WebServer6.0
* Program Code :-

1. **HARDWARE REQUIREMENT:**

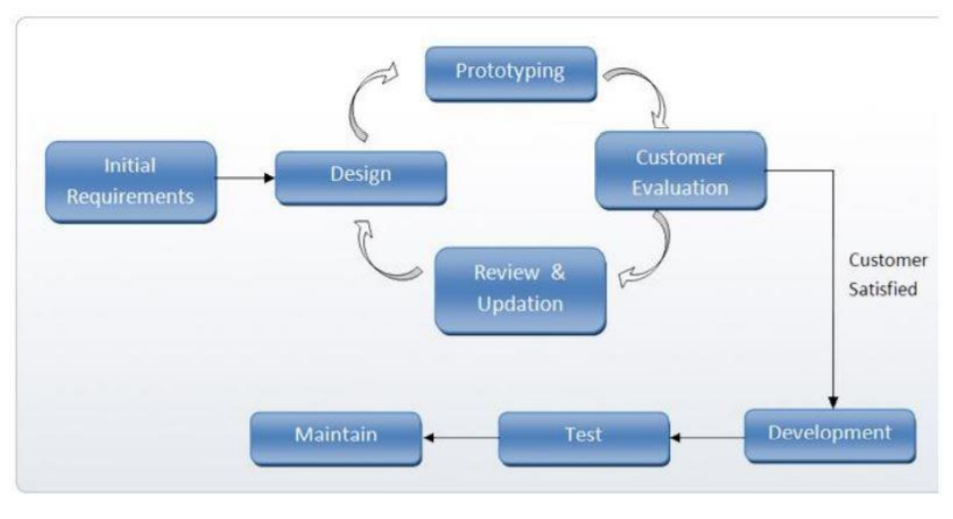
* PIII MHz or above.
* RAM: Minimum 512MB.
* CPUspeed : 2.6GHz.
* Monitor : EGA / SVGA (display), 800X600 24 bits True Color.
* Standard Keyboard :106 Keys with Separate Function Keys & Numeric Pad.
* Mouse : PS /2 Optical mouse.
* CD-ROM : Required

1. **COMMUNICATION INTERFACE:**

* **LAN** (Local Area Network).
* **WAN** (Wide Area Network).
* **VPN** (Virtual Private Network)

The system should perform in such a way that these are to be done to efficiently achieve in the minimum possible time without any errors. The system is designed in such a way that it responds effectively to user’s need.

**Prototype Model**

****

**CONTENTS OF PROPOSED SYSTEM:**

The development of the new system contains the following activities which try to automate the entire process keeping in view of the database integration approach.

1. The Administrators have grates accessibility in collecting the message information that is very much necessary for the system to exist a coordinate and admin has the authorization to provide userid and password to employer. The job of the admin is as follows

* Can login
* Change password
* Approval Company
* Check Company details
* Manage candidates
* Check Student details
* Update Student’s status
* Manage jobs
* Manage recruiters
* Employer status block
* Delete Students and Company
* View reports
* Post jobs
* Notify Students
* Check Placed and Non-placed students
* Search who Placed and non-placed students

2. The Recruiter has grates accessibility in recruiting the candidates for the required post. The job of the Recruiter is as follows

* Register in College
* Edit profile
* Manage job
* Receive candidates
* Change password

3. The Students have the greatest accessibility in searching and applying jobs.

* Can login
* Change password
* Edit profile
* Search jobs
* Check Company availability
* Check Placed or Not
* View Detail of Company’s
* Apply job

**GUI’S**

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI’S at the top level have been categorized

* Administration Login
* Recruiter Login
* Student Login

**Administrator Login:** Admin can contact with the company & can ask the detail information about company

**Recruiter Login:** Recruiter can send message after login.

**Student Login:** When one new employer appointed in the organization, the admin gives him a code.

FUNCTIONAL REQUIREMENTS

# **Functional Requirements of E-RECRUITMENT**

**DESCRIPTION:** The whole system is divided in to four sections like:

* ADMIN SECTION
* RECRUITER SECTION
* STUDENT SECTION
* VISITOR SECTION

**ADMIN SECTION:** This module can control all the requirement of recruiter, student and posted jobs.

**RECRUITER SECTION:** A company can register and login to this site. It can also edit its profile, notify any new jobs to the admin and receive selected candidates.

**STUDENT SECTION:** Students can register to this site then he or she can login to this site, forward his or her C.V, edit his or her profile and apply posted job.

**VISITOR SECTION:** Anyone can visit to this site, search recent job, view updated news and also can give feedback if any.

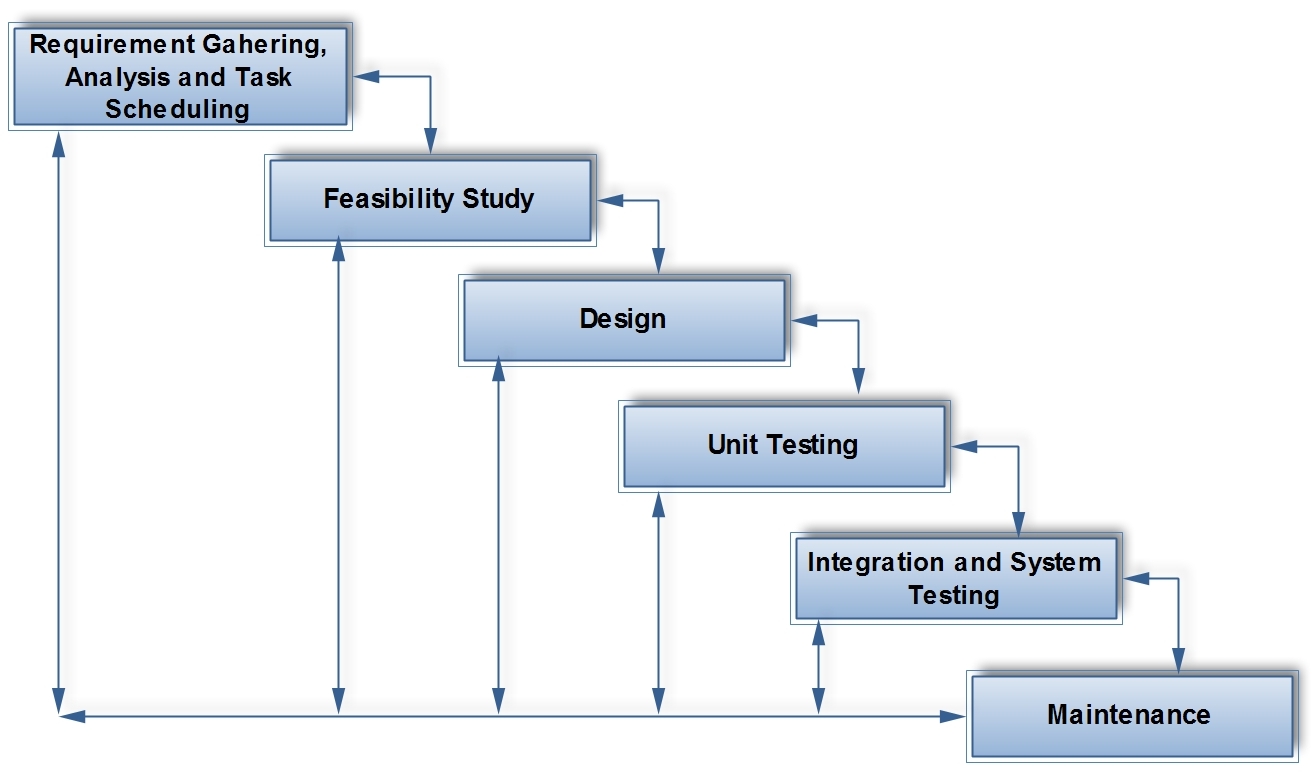
**FUNCTIONAL REQUIREMENT**

* **Authentication**
* Login- The user can login to the PMS system with his/her username and password.
* Logout- The user can log out from the PMS system.
* Login failure- If the user does not exist in the database or the user has not yet being authorized by the PMS admin.
* **Authorization**
* User role check- After logging in, the user role will be checked from the database and the user interface will be displayed according to their role.
* **Process Data**
* Display- User with defined roles can display the content of the database. Being more specific, student can only view his/her personal information. Admin can not only see his/her personal information but also student’s information who are under his/her department or school along with Company’s details. Recruiter can his/her personal details along with the details of the received candidates. Admin and HOD can display their personal information and all students’ information.
* Edit- A user with student role can edit his/her specific personal information. Dean or HOD can only edit students’ personal information that is under his/her coverage except user role type. Admin can edit all information related to all students’ including their user role type.
* Search- User with Dean/HOD role can search the content of database for the students’ who are under his/her coverage. Admin role can search all the students’ information in the database. Search feature works on specific keywords showing student’s enrollment numbers, percentage, departments, qualifications, extra skills, and etc. For example, Admin wants to find students’ who are from “MCA” or “MTECH”. He/she will write the specific keyword in the search bar and press the available search button. Afterwards, he/she will find a list of all the students’ who know “MCA” or “MTECH”. The Students can search from the forum whether they are placed or not after Recruiters post the Selected Candidates. Update authentication- This feature can be used only by admin role type. Admin can update the role type of a specific user. For example, a student passed from the college and his role type will be changed from student role id to Alumni role. Admin will be able to update this authentication mechanism.
* **Recruitment**
* Add new employee- Recruiter role type is able to add a new employee to the database. The new employee will have all the required personal information related to him/her. The new created employee will have an id.
* Update Authentication- After a new employee has being created by Recruiter role, admin role is responsible for updating the students’ information by the specified id assigned in the “Add a new employee” feature. The unique id will be given by the system. Admin will assign a new role such as Alumni, HOD, and admin to the new created user.
* **Report generation**
* Report generation- HOD shall be able to generate a report in pdf format for each student’s information based on the information in the database.
* Recruiter shall be about to generate a report in pdf format for the new employees’ placed in their company from the respective colleges.

**Software Engineering Paradigm Applied**

Software life cycle is the series of identifiable stages that a software product undergoes during its lifetime i.e. a description of software in a systematic and disciplined manner. The life cycle model used to develop this project is the iterative waterfall model. This model emphasizes practical development environment, where the programmers usually commit number of errors in almost every phase of the life cycle model. The various reasons for developing an iterative waterfall model:

* Firstly this method is used to provide feedback paths needed for every phases to its preceding phase to allow for the correctness of the error committed during the phases that are detected in the later phase.
* Secondly the iterative waterfall model breaks down the life cycle into different phases that enhance efficiency of project development.

Thus the nature and the extent of the iteration used in the project are well resembled with the structure of the project. The various developments phases of the iterative waterfall model are shown below:

**Project Planning**

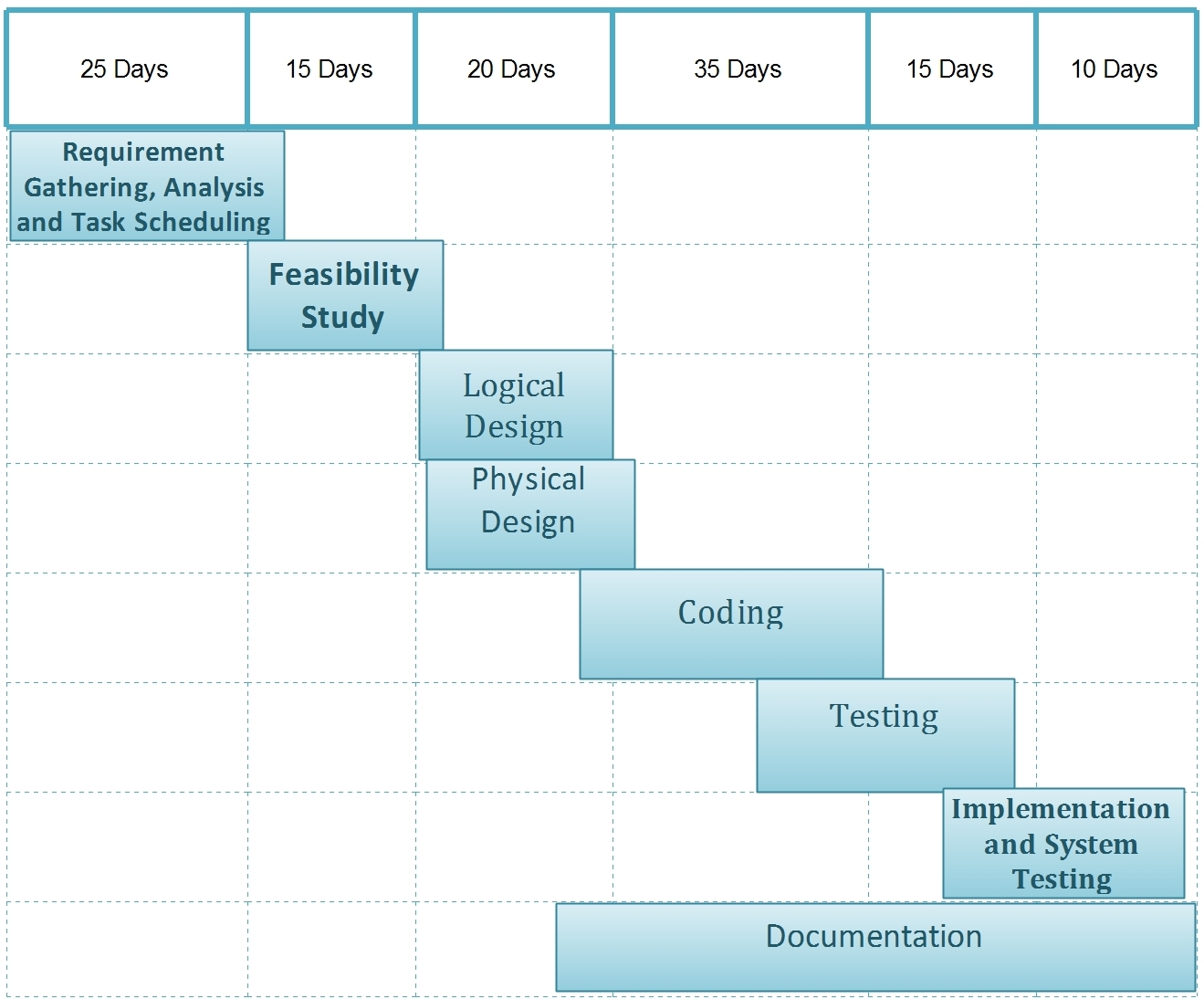
Planning for information systems has a time horizon and a focus dimension. The time horizon dimension specifies the time range of the plan, where as the focus dimension relates whether the primary concern is strategic, managerial, or operational. The system i.e. The Project that we were assigned was required to complete within 17 weeks. What we had planned is as follows:

Requirements analysis, Preliminary Investigation & Information Gathering should be covered within the 1st and 2nd week.

It took 1 week for us to adjust with the tools we have planned to use. It took 8 Weeks for the designing and coding of the system under development and 2 week for Testing & Implementation.

And rest 3 reserve weeks documentation and user testing for any maintenance.

**Project Scheduling**

The work on the proposed system was started on first week of January 2010 and it was estimated to be over by. The following Gantt chart has explained the estimated duration of the different phases of the system development work diagrammatically.

**Gantt chart**

**ER DIAGRAM:**

Conceptual modeling is very important phase in designing a successful database application. Generally the term database application refers to a particular database and the associated program that implement the database queries and updates.

Entity – Relationship model, which is a popular high level conceptual data model. The diagrammatic notations associated with the ER Model are known as ER Diagrams.

The ER model describes data as Entities, Relationships and attributes. The notations used in ER diagram are shown below:

**Meaning** **Symbol**

**Entity (Users)**



**Relationship**



**Attribute**

**Weak Entity**



**Weak Entity Relationship**

 **Derived Attribute**

 **Composite Attribute**



**Total Participation of E2 in R**



**Cardinality Ratio 1: N**



**Structural Constraints (min, max)**

E-R Diagram:

ADMIN

ASSIGN USER ROLE

Generates reports

EMAIL-ID

ADDRESS

EMAIL-ID

BRANCH

BATCH

COMP NAME

PWD

ADMID

PWD

RECID

PWD

ADDRESS

NAME

STDNO

MANAGES

MANAGES

IS A

RECRUITER

COMPANY

STUDENT

PLACEMENT OFFICER/HOD

# **DATA FLOW DIAGRAM (DFD)**

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

**CONTEXT DIAGRAM:**

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

**TYPES OF DATA FLOW DIAGRAMS:**

Data Flow Diagrams are of two types as follows:

* + - Physical DFD
    - Logical DFD

**PHYSICAL DFD:**

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

**LOGICAL DFD:**

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

**BASIC NOTATION:**

The Basic Notation used to create a DFD’s are as follows:

**DATAFLOW:**

Data move in a specific direction from an origin to a destination.

**PROCESS**

People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

**SOURCE:**

External sources or destination of data, which may be People, programs, organizations or other entities.

**DATA STORE:**

Here data are stored or referenced by a process in

theSystem

**DESIGN:**

Design is the first step in moving from problem domain to the solution domain. Design is essentially the bridge between requirements specification and the final solution.

The goal of design process is to produce a model or representation of a system, which can be used later to build that system. The produced model is called the “Design of the System”. It is a plan for a solution for the system.

**Constructing a DFD:**

Several rules of thumb are used in drawing DFD’s:

Process should be named and numbered for an easy reference. Each name should be representative of the process.

The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

When a process is exploded into lower level details, they are numbered.

The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out. Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

**SAILENT FEATURES OF DFD’s**

1. The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
3. The sequence of events is not brought out on the DFD.

Rules Governing the DFDs

PROCESS

1. No process can have only outputs.
2. No process can have only inputs. If an object has only inputs than it must be a sink.
3. A process has a verb phrase label.

DATA STORE

1. Data cannot move directly from one data store to another data store, a process must move data.
2. Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into data store
3. A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data

1. Data cannot move direly from a source to sink it must be moved by a process
2. A source and /or sink has a noun phrase land

DATA FLOW

1. A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated however by two separate arrows since these happen at different type.
2. A join in DFD means that exactly the same data comes from any of two or more different processes data store or sink to a common location.
3. A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produce some other data flow returns the original data into the beginning process.
4. A Data flow to a data store means update (delete or change).
5. A data Flow from a data store means retrieve or use.

A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow as long as all of the flows on the same arrow move together as one package.

**DATA FLOW DIAGRAM** :

**CONTEXT**

Student Detail

Student

Job List

Request for Data

Post Job

Campus

Job process details

Company

Get ACK.

Edit details

Admin

Database

**LOGIN FORM PROCES OF DFD FOR STUDENT :**

Login Information

Log in

Process

ACK.

Resister

User

Insert New Info.

Student Information

Student

**LOGIN FORM PROCES OF DFD FOR COMPANY:**

Login

Information

Login

Process

Request

ACK.

Get Info.

Invalid

Register

User

Insert New Info.

Company Information

Company

**LOGIN FORM PROCES OF DFD FOR ADMIN:**

Login Info. Request Info.

Login

Get Info**.**

ACK

Admin Info.

Admin

**JOB (RELETED) PROCESS OF STUDENT :**

Job process

Comp.

Comp List

Student

ACK.

Apply for Job

Student

If Qualified

Receive Alert

* + 1. **COMPANY VIEW PROCESS OF STUDENTS :**

Student List

Students

List

ACK**.**

Select

Candidate

Send

Alert

Student Details

Registration Fill

Company List

Company

**ADMIN VIEW PROCESS FOR ALL STUDENTS AND COMPANES**

Insert/Update/Delete

Students

Information

View Placed or not

Maintenance

Admin

ACK.

Job Process

Insert/Update/Delete

Company’s Info

**Searching Process Diagram**

Login

Registration

View Registration

Companies

View Registration

Students

Student

Company

Demands of

Company

View Placed or

Not

Search

Companies.

**Use case diagram**

Login

Registration

Student Details

Company Details

**Students**

Placed or not

**Admin**

Student List

Company List

Decide Salary

**Companies**

### **Process Flow Diagram**

A process flow diagram is pictorial representation of algorithm, as it represents solution in from of picture. It is easier to understand and develop. A main advantage of flow chart is visibility of paths within solution, each path is clearly visible as arrows are used to represents flow.

Yes

Want to

edit resume?

No

Search job

Edit/update

Login



Yes

Student

registered or not?

No

No

Company

registered or not

Yes

Yes

Edit/up

Year Jobs

No

Start

Check Student list

Edit update List branch std

Login

Login

Register

Register

Apply

Select candidate

Stop

Select Comp

# **Sequence Diagram**

Create Login Submit Resume

Create Login

Submit Requirement

Match up Resume with Requirement

Send Alert

Send Alert

Maintenance

Success Story

Success Story

Approved Story

Administrator

Server

Company

Student

**USE CASE DIAGRAMS:**

**UNIFIED MODELING LANGUAGE**

UML is the international standard notation for object-oriented analysis and design. The Object Management Group defines it. The heart of object-oriented problem solving is the construction of a model. The model abstracts the essential details of the underlying problem from its usually complicated real world. Several modeling tools are wrapped under the heading of the **UML**™, which stands for Unified Modeling Language™.

**AN OVERVIEW OF UML:**

The UML is a language for

* Visualizing
* Specifying
* Constructing
* Documenting

These are the artifacts of a software-intensive system. The three major elements of UML are

* The UML’s basic building blocks
* The rules that dictate how those building blocks may be put together.
* Some common mechanisms that apply throughout the UML.

# 

# BASIC BUILDING BLOCKS OF THE UML:

The vocabulary of UML encompasses three kinds of building blocks:

* Things
* Relationships
* Diagrams

**Things** are the abstractions that are first-class citizens in a model.

**Relationships** tie these things together.

**Diagrams** group the interesting collection of things.

**THINGS IN THE UML:**

They are the abstractions that are first-class citizens in a model. There are four kinds of things in the UML

* Structural things
* Behavioral things.
* Grouping things.
* Annotational things.

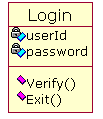
These things are the basic object oriented building blocks of the UML. They are used to write well-formed models.

**STRUCTURAL THINGS:**

Structural things are the nouns of the UML models. These are mostly static parts of the model, representing elements that are either conceptual or physical. In all, there are seven kinds of Structural things.

**CLASS:**

A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces. Graphically a class is rendered as a rectangle, usually including its name, attributes and operations, as shown below.



**INTERFACE:**

An interface is a collection of operations that specify a service of a class or component. An interface describes the externally visible behavior of that element.

Graphically the interface is rendered as a circle together with its name.

**COLLABORATION:**

Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that’s bigger than the sum of all the elements. Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name as shown below.

Chain

**USE CASE:**

Use case is a description of a set of sequence of actions that a system performs that yields an observable result of value to a particular thing in a model. Graphically, Use Case is rendered as an ellipse with dashed lines, usually including only its name as shown below.

**ACTIVE CLASS:**

An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity. Graphically, an active class is rendered just like a class, but with heavy lines usually including its name, attributes and operations as shown below.

HRMS

EMPLOYEE

DEATILS

Suspend ()

Flush ()

**COMPONENT:**

Component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs, usually including only its name, as shown below.



**NODE:**

A Node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and often, processing capability. Graphically, a node is rendered as a cube, usually including only its name, as shown below.



**BEHAVIORAL THINGS:**

Behavioral Things are the dynamic parts of UML models. These are the verbs of a model, representing behavior over time and space.

**INTERACTION:**

An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose. Graphically, a message is rendered as a direct line, almost always including the name if its operation, as shown below.

Display

**STATE MACHINE:**

A state machine is a behavior that specifies the sequence of states an object are an interaction goes through during its lifetime on response to events, together with its responses to those events. Graphically, a state is rendered as a rounded rectangle usually including its name and its sub-states, if any, as shown below.

Waiting

**GROUPING THINGS:**

Grouping things are the organizational parts of the UML models. These are the boxes into which a model can be decomposed.

**PACKAGE:**

A package is a general-purpose mechanism for organizing elements into groups.



**ANNOTATIONAL THINGS:**

Annotational things are the explanatory parts of the UML models.

**Note:**

A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements.

Graphically a note is rendered as a rectangle with dog-eared corner together, with a textual or graphical comment, as shown below.



**RELATIONSHIPS IN THE UML:**

There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

**1. DEPENDENCY:**

This is relationship between two classes whenever one class is completely dependent on the other class. Graphically the dashed line represents it with arrow pointing to the class that it is being depended on.



**2. ASSOCIATION:** It is a relationship between instances of the two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work. In a diagram, an association is a link connecting two classes. Graphically it is represented by line as shown.

**3. GENERALIZATION:**

An inheritance is a link indicating one class is a super class of the other. A generalization has a triangle pointing to the super class. Graphically it is represented by line with a triangle at end as shown.

**4. REALIZATION:**

**DIAGRAMS IN UML:**

Diagrams play a very important role in the UML. There are nine kind of modeling diagrams as follows:

* Use Case Diagram
* Class Diagram
* Object Diagram
* Sequence Diagram
* Collaboration Diagram
* State Chart Diagram
* Activity Diagram
* Component Diagram
* Deployment Diagram

## CLASS DIAGRAM:

Class diagrams are the most common diagrams found in modeling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Graphically, a class diagram is a collection of vertices and arcs.

**Contents:**

Class Diagrams commonly contain the following things:

Classes

Interfaces

Collaborations

Dependency, generalization and association relationships

#### **USE CASES DIAGRAM:**

Use Case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of systems (activity diagrams, sequence diagrams, state chart diagrams and collaboration diagrams are the four other kinds of diagrams in the UML for modeling the dynamic aspects of systems). Use Case diagrams are central to modeling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relationships.

**COMMON PROPERTIES:**

A Use Case diagram is just a special kind of diagram and shares the same common properties, as do all other diagrams- a name and graphical contents that are a projection into the model. What distinguishes a use case diagram from all other kinds of diagrams is its particular content.

**Contents**

Use Case diagrams commonly contain:

Use Cases

Actors

Dependency, generalization, and association relationships

Like all other diagrams, use case diagrams may contain notes and constraints. Use Case diagrams may also contain packages, which are used to group elements of your model into larger chunks. Occasionally, you will want to place instances of use cases in your diagrams, as well, especially when you want to visualize a specific executing system.

#### **INTERACTION DIAGRAMS:**

An Interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. Interaction diagrams are used for modeling the dynamic aspects of the system.

A sequence diagram is an interaction diagram that emphasizes the time ordering of the messages. Graphically, a sequence diagram is a table that shows objects arranged along the X-axis and messages, ordered in increasing time, along the Y-axis and messages, ordered in increasing time, along the Y-axis.

**Contents**

Interaction diagrams commonly contain:

Objects

Links

Messages

Like all other diagrams, interaction diagrams may contain notes and constraints.

**SEQUENCE DIAGRAMS**:

A sequence diagram is an interaction diagram that emphasizes the time ordering of the messages. Graphically, a sequence diagram is a table that shows objects arranged along the X-axis and messages, ordered in increasing time, along the Y-axis.

Typically you place the object that initiates the interaction at the left and increasingly more sub-routine objects to the right. Next, you place the messages that these objects send and receive along the Y-axis, in order of increasing time from top to the bottom. This gives the reader a clear visual cue to the flow of control over time.

**Sequence diagrams have two interesting features**:

1. There is the object lifeline. An object lifeline is the vertical dashed line that represents the existence of an object over a period of time. Most objects that appear in the interaction diagrams will be in existence for the duration of the interaction, so these objects are all aligned at the top of the diagram, with their lifelines drawn from the top of the diagram to the bottom.
2. There is a focus of the control. The focus of control is tall, thin rectangle that shows the period of time during which an object is performing an action, either directly or through the subordinate procedure. The top of the rectangle is aligns with the action; the bottom is aligned with its completion.
3. **Contents**

Sequence diagrams commonly contains

Objects

Object Life Line

Focus of Control

#### **ACTIVITY DIAGRAM**

An Activity Diagram is essentially a flow chart showing flow of control from activity to activity. They are used to model the dynamic aspects of as system. They can also be used to model the flow of an object as it moves from state to state at different points in the flow of control.

An activity is an ongoing non-atomic execution with in a

State machine. Activities ultimately result in some action, which is made up of executable atomic computations that result in a change of state of distinguishes a use case diagram from all other kinds of diagrams is its particular content.

**Contents**

Activity diagrams commonly contain:

Fork

Start & End Symbol

#### STATE CHART DIAGRAMS

A state chart diagram shows a state machine. State chart diagrams are used to model the dynamic aspects of the system. For the most part this involves modeling the behavior of the reactive objects. A reactive object is one whose behavior is best characterized by its response to events dispatched from outside its context. A reactive object has a clear lifeline whose current behavior is affected by its past.

A state chart diagram show a state machine emphasizing the flow of control from state to state. A state machine is a behavior that specifies the sequence of states an object goes through during its lifetime in response to events together with its

Response to those events. A state is a condition in the life of the object during which it satisfies some conditions, performs some activity or wait for some events. An event is a specification of a significant occurrence that has a location in time and space.

Graphically a state chart diagram is a collection of vertices and arcs.

**COLLABORATION DIAGRAM:**

A collaboration diagram shows both the structural and the behavioral aspects explicitly. This is unlike a sequence diagram which shows only the behavioral aspects. The structural aspect of a collaboration diagram consists of objects and the links existing between them. In this diagram an object is also called a collaborator. The behavioral aspect is described by the set of messages exchanged among the different collaborators.

The link between objects is shown as a solid line and can be used to send messages between two objects. The message is shown as labeled arrow placed in near the link. Messages are prefixed with sequence numbers because that is the only way to describe the relative sequencing of the messages in this diagram. The collaboration diagram is shown below.

**SEQUENCE DIAGRAM:**

A sequence diagram shows interaction among objects as a two-dimensionalchart. The chart is read from top to bottom. The objects participating in the interaction are shown at the top of the chart as boxes attached to a vertical dashed line. Inside the box the name of the object is written with a colon separating it from the name of the class, and both the name of the object and the class are underlined.

The object appearing at the top signifies that the object already existed when the use case execution was initiated. However, if some object is created during the execution of the use case and participate in the interaction, then the object should be shown at the appropriate place on the diagram where it was created

**Table Structure:**

1. **USER REGISTRATION :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELD | DATA TYPES | CONSTRAINTS | KEYS | Description |
| Uid | numeric(18, 0) | Not null |  | User ID |
| Uname | varchar(20) | Not null | Pk | User Name |
| Pwd | varchar(20) | Not null |  | Password of User |
| Cpwd | varchar(20) | Not null |  | Confirm Password of User |
| Urole | varchar(10) | Not null |  | Role of New User |

2**.(Admin\_Login\_Master)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELD | DATATYPE | CONSTRAINT | KEY | DESCRIPTION |
| admin\_id | Int | Not null | Pk | unique identification |
| admin\_userid | varchar(50) | Not null |  | Login\_user\_id |
| Admin\_pwd | Varchar(50) | Not null |  | Administrator password |
| Admin\_email | Varchar(50) | Not null |  | Admin\_email\_Id |

3. **(Recruiter\_Login\_Master)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELD | DATATYPE | CONSTRAINT | KEY | DESCRIPTION |
| rec\_id | Int | Not null | Pk | unique identification |
| rec\_userid | varchar(50) | Not null |  | Login\_user\_id |
| rec\_pwd | Varchar(50) | Not null |  | Recruiter password |
| rec\_email | Varchar(50) | Not null |  | Recruiter\_email\_Id |

4**.(Recruiter\_ master)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINT** | **KEY** | **DESCRIPTION** |
| rec\_id | Int | notnull | Pk | unique identification |
| rec\_userid | varchar(50) | notnull |  | recruiter user id |
| rec\_pwd | varchar(50) | notnull |  | recruiter password |
| rec\_name | varchar(50) | notnull |  | name of the company |
| rec\_emailid | varchar(50) | notnull |  | email id of recruiter |
| rec\_mobilno | varchar(50) | notnull |  | mobile number of recruiter |
| Industry | varchar(50) | notnull |  | type of industry |
| company\_desc | varchar(100) | notnull |  | desc. of company |
| Address | varchar(50) | notnull |  | address. of company |
| rec\_status | varchar(50) | notnull |  | active/inactive |
| rec\_website | varchar(50) | notnull |  | website of employer |

**4. STUDENTS REGISTRATION :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELD | DATA TYPES | CONSTRAINTS | KEYS | Description |
| stdid | numeric(18, 0) | Not null | - | Student ID Number |
| stderno | numeric(20, 0) | Not null | Primary Key | Student Enrollment Number |
| fname | varchar(20) | Not null | - | Student First Name |
| lname | varchar(20) | Not null | - | Student Last Name |
| preadd | varchar(30) | Not null | - | Present address of Student |
| paradd | varchar(30) | Not null | - | Student Permanent address |
| dob | date time | Not null | - | Student Date Of Birth |
| city | varchar(10) | Not null | - | City of Student |
| pincode | numeric(6, 0) | Not null | - | Student City Pin code |
| emailid | varchar(30) | Not null | - | Email-ID of Student |
| gender | varchar(10) | Not null | - | Student = Male or Female |
| branch | varchar(20) | Not null | - | Branch of Student |
| Extra course | varchar(50) | Not null |  | Selected Extra Course |

5. (Student\_Login\_Master)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIELD | DATATYPE | CONSTRAINT | KEY | DESCRIPTION |
| stderno | Int | Not null | Pk | unique identification |
| std\_userid | varchar(50) | Not null |  | Login\_user\_id |
| std\_pwd | Varchar(50) | Not null |  | Student password |
| email\_id | Varchar(50) | Not null |  | Recruiter\_email\_Id |

6. (Student\_contact\_master)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINT** | **KEYS** | **DESCRIPTION** |
| email\_id | varchar(50) | Not null | Pk | Email Id of candidate |
| State | varchar(50) | Not null |  | state of candidate |
| City | varchar(50) | Not null |  | city of candidate |
| contact no | varchar(50) | Null |  | contact no. of candidate |
| mobile no. | varchar(50) | Null |  | mobile no. of candidate |

8. (Student\_edu\_master)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINT** | **KEYS** | **DESCRIPTION** |
| email\_id | varchar(50) | null | - | email id of the candidate |
| ug\_course\_name | varchar(50) | null | - | ug course name |
| ug\_college | varchar(50) | null | - | ug college name |
| ug\_university | varchar(50) | null | - | ug universityname |
| ug\_pass\_year | Int | null | - | year of passing |
| ug\_percentage | varchar(50) | null | - | Percentage |
| bg\_course\_name | varchar(50) | null | - | bg course name |
| bg\_college | varchar(50) | null | - | college name |
| bg\_university | varchar(50) | null | - | bg universityname |
| bg\_pass\_year | Int | null | - | year of passing |
| bg\_percentage | varchar(50) | null | - | Percentage |
| pg\_course\_name | varchar(50) | null | - | pg course name |
| pg\_college | varchar(50) | null | - | college name |
| pg\_university | varchar(50) | null | - | pg universityname |
| pg\_pass\_year | Int | null | - | year of passing |
| pg\_percentage | varchar(50) | null | - | Percentage |
| certificate | varchar(50) | null | - | certified form which industry |

9. Pin-code and City :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA TYPES** | **CONSTRAINTS** | **KEYS** | **DESCRIPTION** |
| id | numeric(18,0) | Not null | - | Table Record-Id |
| cityname | varchar(50) | Not null | - | City Name |
| pincode | numeric(18) | Not null | Primary key | Pin code Of City |

10. Placement (Job Accesses):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** | **KEYS** | **DESCRIPTION** |
| Pid | numeric(18,0) | Not null | - | Table Record-Id |
| Erno | numeric(18,0) | Not null | Primary Key | Student Enrollment No |
| stdname | varchar(50) | Not null | - | Student Name |
| Batch | numeric(15,0) | Not null | - | Student Access Job Year |
| compname | varchar(50) | Not null | - | Name of Company |
| branch | varchar(50) | Not null | - | Branch of Student |
| compcity | varchar(50) | Not null | - | City of Company |
| compadd | varchar(50) | Not null | - | Address of Company |
| Salary | numeric(18,0) | Not null | - | Salary of Student |
| compcontactno | numeric(18,0) | Not null | - | Company Contact Number |
| compeid | varchar(50) | Not null | - | Email-id of Company |
| compwebadd | varchar(50) | Not null | - | Web Address of Company |
| comphrname | varchar(50) | Not null | - | HR Manager Name of  Company |

11.( Apply\_job\_master)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **FIELD** | **CONSTRAINT** | **KEYS** | **DESCRIPTION** |
| email\_id | Int | Not null |  | email\_id of candidate |
| job\_id | varchar(50) | Not null |  | job id candidate |
| job\_title | varchar(50) | Not null |  | title of job |
| job\_desc | varchar(50) | Not null |  | describtion of job |
| apply\_status | varchar(50) | Not null |  | status of apply |
| companyname | varchar(50) | Not null |  | name of company |

12.( Feedback\_master)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD** | **FIELD** | **CONSTRAINT** | **KEYS** | **DESCRIPTION** |
| name | Varchar(50) | notnull |  | name of candidate |
| Email\_id | varchar(50) | notnull |  | email\_id of candidate |
| subject | varchar(50) | notnull |  | Heading of feedback to be given |
| feedback | varchar(50) | notnull |  | describtion of subject want to post |

**PROCESS LOGIC:**

**DBFUNCTION:**

**INSERTDATA:** The objective of this function is to insert data in to the table (emp\_master1).It contains two arguments: table name as tbname and value as val.

Syntax: Insert data(string tbname, string val)

{

}

**INSERTDATAWITHSPECIFICCOLUMN:** The objective of this function is to insert data within a specific column in a table. It contains three arguments: table name, column name and value.

Syntax: insertdatacolm(string tbname, string colname, string val)

{

}

**DISPLAYGRID:** The objective of this function is to display all the data of table(emp\_master1)in Grid view control. It contains two arguments : tbname and a grid view object as grd.

Syntax: Displaygrid( string tbname, Grid view grd)

{

}

**DISPLAYGRIDWITHCOND:** This function display data in a Grid view control according to condition given by user. It contains four arguments :tbname ,condition as cond, value as condval and Gridview object as grd.

Syntax: displaygridwithcond(string tbname, string cond, string condval, Gridview grd)

{

}

**POPULATEDROPDOWN:**

This function is used to display the selected data from dropdownlist in Gridview. It contains two arguments : tablename and Dropdownlist.

Syntax:populatedrop(string tbname, string disval, string hidval, DropDownList drp)

{

}

# IMPLEMENTATION METHODOLOGY:

**POPULATEDROPDOWNWITHCONDITION:** This function is used to display the data in Gridview according to user condition. It contains four arguments : table name, condition , condition value and Gridview.

Syntax: populatedropwithcond(string tbname, string disval, string hidval, string cond, string condval, DropDownList drp)

{

}

**RETURN DATASET:** This function is used to return dataset.It contains three arguments : table name, condition and conditionvalue.

Syntax: DataSet returndataset(string tbname, string cond, string condval)

{

}

**DELETEDATA:** This function is used to delete the data from table. It contains two arguments : table name and column name.

Syntax: deletedata(string tbname, string colname)

{

}

Implementation

&

Testing

1. **Implementation and Testing**

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the “WHY” and “HOW” of product validation. It should be through enough to be useful but not so thorough that no one outside the test group will read it.

* 1. **Test cases**

A test case is a document that describes an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test case name, objective, test condition, input data requirements expected results. The process of developing test cases can help find problems in the requirements or design of an application, since it requires completely thinking through the operation of the application. For this reason, it’s useful to prepare test cases early in development cycle if possible.

* 1. **Goal of testing**

“Program testing can be used to show the presence of bug, but never to show their absence.” If the results delivered by the system are different from the expected ones then the system is incorrect and these bugs should be fixed.

* 1. **Test units**

In this project two stage of testing are to be performed, namely unit testing and integrating testing. All the basic units mentioned in the design document are to be tested.

**Features to be tested:**

* All the functional features in the analysis documents are to be tested.
* Every intermediate result of each module is matched with the expected result.
  1. **Integration test report**

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

* Whether the software matches with the specification (i.e. process base) to verify the product
* Whether this software is one what user wants (i.e. product base) to validate the product

Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behavior if exists. Later all the module was integrated and flat file was generated.

Here are basic function units of the system were integrated and the testing was done to see if there were any erroneous results. During integration testing some errors were detected out of which most of them are fixed.

The test type with the results of some of the tests is shown below:

**Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step #** | **Step Details** | **Expected Inputs** | **Expected Results** | **Pass / Fail / Not executed / Suspended** |
|  |
| 1 | Navigate to http://placementmanagement system.in | Site should Open | As Expected |  |  |
| 2 | Enter on Student Registration | Student Regd page should be shown | As Expected |  |  |
| 3 | Enter First Name | Student can Enter both Uppercase letters and lower case letters | numeric data and,special characters and blank spaces not allowed |  |  |
| 4 | Enter last Name | Student can Enter both Uppercase letters and lower case letters | numeric data and,special characters and blank spaces not allowed |  |  |
| 5 | Enter Dob | Should enter numeric,special character and blank space | Dob should not be blank and blank spaces are not allowed |  |  |
| 6 | Enter Mobile Number | student enter 9digits number with special characters | should enter only numeric values and it should be of 10 digits numbers |  |  |
| 7 | Enter Present Address | Student enter uppercase/lowercase letters,numerics | should not be blank and no special characters are allowed |  |  |
| 8 | Enter Permanent Address | Student enter uppercase/lowercase letters,numerics | should not be blank and no special characters are allowed |  |  |
| 10 | Enter pincode | Student Enter 8 digits number and letters | should be strictly of 6 digits and only numerics are allowed |  |  |
| 12 | Enter User Id | student enter user id/email by using blank space uppercase,lowercase letters | user id should not be blank nad blank spaces are not allowed |  |  |
| 13 | Enter Password | Student enter uppercase/lowercase letters,numerics and special characters | password should not be blank and should be a mix of letters,numerics and special characters otherwise not accepted |  |  |

**Student Registration**

**Admin Registration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step #** | **Step Details** | **Expected Inputs** | **Expected Results** | **Pass / Fail / Not executed / Suspended** |
|  |
| 1 | Navigate to http://placementmanagement system.in | Site should Open | As Expected |  |  |
| 2 | Enter on Admin Registration | Admin Regd page should be shown | As Expected |  |  |
| 3 | Enter First Name | Admin can Enter both Uppercase letters and lower case letters | numeric data and,special characters and blank spaces not allowed |  |  |
| 4 | Enter last Name | Admin can Enter both Uppercase letters and lower case letters | numeric data and,special characters and blank spaces not allowed |  |  |
| 5 | Enter User Id | Admin enter user id/email by using blank space uppercase,lowercase letters | user id should not be blank nad blank spaces are not allowed |  |  |
| 6 | Enter Password | Admin enter uppercase/lowercase letters,numerics and special characters | password should not be blank and should be a mix of letters,numerics and special characters otherwise not accepted |  |  |
| 6 | Click Register | User is registered Successfully | Userid and Password sent to database |  |  |
| 6 | Click Login | Customer is Logged In | as Excepted |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step #** | **Step Details** | **Expected Inputs** | **Expected Results** | **Pass / Fail / Not executed / Suspended** |
|  |
| 1 | Navigate to http://placementmanagement system.in | Site should Open | As Expected |  |  |
| 2 | Enter on Company Registration | recruiter registration page should open | As Expected |  |  |
| 3 | Enter Company Name | recruiter can Enter both Uppercase letters and lower case letters | Comp Name should not be blank and numerics are not allowed |  |  |
| 4 | Enter Branch Name | Recruiter can Enter the branch name | if not recognised as a verified place show erroe message |  |  |
| 5 | Enter Mobile Number | recruiter enter 9digits number with special characters | should enter only numeric values and it should be of 10 digits numbers |  |  |
| 6 | Enter Present Address | recruiter enter uppercase/lowercase letters,numerics | should not be blank and no special characters are allowed |  |  |
| 7 | Enter Data for Post Job | Recruiter check for updating Status in Post Job | if Status not changed,error message should be shown that "Sorry! Invalid Status |  |  |
| 8 | Enter Login Input Data | Home page should be shown | as Expected |  |  |
| 9 | Enter User Id | recruiter enter user id/email by using blank space uppercase,lowercase letters | user id should not be blank nad blank spaces are not allowed |  |  |
| 10 | Enter Password | recruiter enter uppercase/lowercase letters,numerics and special characters | password should not be blank and should be a mix of letters,numerics and special characters otherwise not accepted |  |  |
| 11 | Click Register | User is registered Successfully | Userid and Password sent to database |  |  |
| 12 | Click Login | Customer is Logged In | as Excepted |  |  |

**Recruiter Module Test Case**

* 1. **Functional testing**

These are the points concerned during the stress test:

* Nominal inputs: character is inputted in the place of digits and the system has to flash the message “Data error”
* Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

**Testing Method Used**

We have adopted a testing method which is a mix of both white box (structural)and black box (functional) testing. For modules we have adopted white box testing. Then we integrated the modules into sub-systems and further into the system. There we adopted black box testing for checking the correctness of the system.

**Requirements validated and verified:**

* The data is getting entered properly into database
* The screens are being loaded correctly.
* The various functions specified are being performed completely.

Cost Estimation

**COST ESTIMATION**

Software Cost Estimation continues to be a weak link in software project management. The aim of this web-site is to present a review of current cost estimation techniques to help both industry and academia in choosing the appropriate methods when preparing software cost estimates. The site covers both traditional and state of the art methods identifying advantages and disadvantages of each and the underlying aspects in preparing cost estimates. The site also provides links to other software cost estimation sites which are involved in this area and details the research that has been undertaken at Bournemouth University. Click on the link icon in the frame to get a comprehensive list of software cost estimation sites on the web.

For an introduction to software cost estimation, click on the link below else choose a SCE method to explore.

INTRODUCTION:

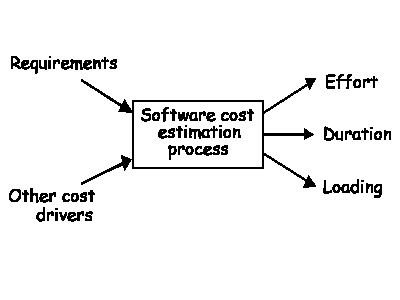
The majority of software cost estimates are based on the following methods:

### ALGORITHMIC MODELS EXPERTS JUDGEMENT MACHINE LEARNING

**ALGORITHMIC COST MODELS**

To date most work carried out in the software cost estimation field has focused on algorithmic cost modeling. In this process costs are analyzed using mathematical formulae linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. The accuracy of the model can be improved by [calibrating](http://www.ecfc.u-net.com/cost/calibrat.htm) the model to your specific development environment, which basically involves adjusting the weightings of the metrics.

#### Classical view of the algorithmic cost estimation process



Algorithmic models generally provide direct estimates of effort or duration. As shown in figure the main input is usually a prediction of software size. Effort prediction models take the general form :

**effort = p\*S**

**GANTT CHART**

It is also called bar chart. A Gantt chart is perhaps the simplest form of formal project management. It is used almost exclusively for scheduling purposes and therefore controls only the time dimension of projects

**Gantt charts (developed by Henry L. Gantt) are a project control technique that can be used for several purposes, including scheduling, budgeting and resource planning. A Gantt chart is a bar chart, with each bar representing an activity. The bars are drawn against a time line. The length of each bar is proportional to the length of time planned for the activity. A Gantt chart helps in scheduling the activities of a project, but it does not help in identifying them. One can begin with the work breakdown structure. During the scheduling activity, and also during implementation of the project, new activities may be identified that were not envisioned during the initial planning. The manager must then go back and revise the breakdown structure and the schedules to deal with these new activities.**

**It is also termed as Time line chart, which depicts, a part of a software project schedule that emphasizes the concept-scooping task for a software product. All project tasks are listed in the left hand column. The horizontal bar indicates the duration of each task. When multiple bars occur at the same time on the calendar, task concurrency is complicated.**

## ADVANTAGES

Gantt charts can take different forms depending on their intended use.

They are best for resource planning and scheduling.

## DISADVANTAGES

A Gantt chart helps in scheduling the activities of a project, but it fails to identify them.

While Gantt chart show the tasks and their duration clearly however, they do not show inter task dependencies plainly.

Gantt chart gives little indication of which tasks must be completed before others are begun.

**PERT CHART**

Program Evaluation Review Technique (PERT) is a project scheduling method that can be applied to software development. This is used for following project-planning activities.

* Estimation of efforts
* Decomposition of product function.
* The selection of appropriate process model.
* The selection of project type and task set.

Unlike bar charts, PERT can be both a cost and a time management system; PERT is organized by events and activities or tasks. PERT has several advantages over bar charts and is likely to be used with more complex projects. One advantage of PERT is that it is a scheduling device that also shows graphically which tasks must be completed before others are begun. Also by displaying the various task paths, PERT enables the calculation of a critical path. Each path consists of combinations of tasks, which must be completed. The time and cost associated with each task along a path are calculated, and the path that requires the greatest amount of elapsed time is the critical path. Calculation of the critical path enables project managers to monitor the series of tasks more closely than others and to shift resources to it if it begins to fall behind schedule.

PERT controls time and costs during the project and also facilitates finding the right balance between completing a project on time and completing it within the budget. PERT recognizes that projects are complex, that some tasks must be completed before others can be started, and that the appropriate way to manage a project is to define and control each task. Because projects often fall behind schedule, PERT is designed to facilitate getting a project back on schedule. PERT is based in part on the premise that subjective estimates of the total completion time for a project are usually greatly inferior to the sum of subjective estimates for each task. As with Gantt charts, to build a PERT chart for a project, one must first list all the activities required for completion of the project and estimate how long each will take. Then one must determine the dependence of the activities on each other. The PERT chart gives a graphical representation of this information. Clearly, the technique does not help in deciding which activities are necessary or how long each will take, but it does force the manager to take the necessary planning steps to answer these questions.

**ADVANTAGES**

* It forces the developer to plan.
* It shows the interrelationships among the tasks in the project and, in particular, clearly identifies the critical path of the project, thus helping to focus on it. The fact that the PERT chart has exposed the critical path allows us the opportunity to consider alternative approaches to cope with a potential problem.
* It exposes all possible parallelism in the activities and thus helps in allocating resources.
* It allows scheduling and simulation of alternative schedules.
* It enables the project manager to monitor and control the project.

**Pert chart for the given project is as follows:**

# Start

# Analysis

# Write Manual

# Build Code Generator

# Design

# Integration

**& Test**

# Finish

**PERT CHART OF alert message system**

LIFE

As said earlier in this document, one of the additions that will enhance the efficacy of the project.

With the addition of advanced test toggling features in the project entire the integrated client will also have better control over the way the engine performs. This interface will also allow the integrated client to search for engine updated and client related implementation updates.