

Payroll management system explain itself that it's an accounting package that will provide a financial solution for a particular organization or for a particular institutional body. This payroll management helps accounts to maintain all records by using predefined rules set by the organization working pattern. Its admin module has been developed in such a pattern that allows to change the working rules any time as per the business requirement. Its automatic rates and salary are fixed which saves processing time for preparing final ledger. Admin will only have to fill the database with only employee's details and select their type of work and their post and final report will be prepared by the generation module. As codes has been developed by using the concepts of object-oriented programming so, coding reusability can easily be achieved. Even if there will be changes to made in future, it can be performed by the technical person within few seconds just by changing the working rules.

PAYROLL



PhD Scholar in Computer Science and Engineering, with Master of Technology in Information Security and Cyber Forensics and Bachelor of Technology in Information Technology. Residing in New Delhi, India.

FOR AUTHOR USE

Mehtab Alam
Iftikhar Ahmed

Payroll Management System

Managing the payment system-With efficiency



Mehtab Alam
Iftikhar Ahmed

Payroll Management System

FOR AUTHOR USE ONLY

**Mehtab Alam
Iftikhar Ahmed**

Payroll Management System

Managing the payment system-With efficiency

FOR AUTHOR USE ONLY

LAP LAMBERT Academic Publishing

Imprint

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Cover image: www.ingimage.com

Publisher:

LAP LAMBERT Academic Publishing

is a trademark of

Dodo Books Indian Ocean Ltd., member of the OmniScriptum S.R.L Publishing group

str. A.Russo 15, of. 61, Chisinau-2068, Republic of Moldova Europe

Printed at: see last page

ISBN: 978-620-3-86260-7

Copyright © Mehtab Alam, Iftikhar Ahmed

Copyright © 2021 Dodo Books Indian Ocean Ltd., member of the OmniScriptum S.R.L Publishing group

FOR AUTHOR USE ONLY

PAYROLL MANAGEMENT

Payroll management system explain itself that it's an accounting package that will provide a financial solution for a particular organization or for a particular institutional body. This payroll management helps accounts to maintain all records by using predefined rules set by the organization working pattern.

Its admin module has been developed in such a pattern that allows to change the working rules any time as per the business requirement. Its automatic rates and salary are fixed which saves processing time for preparing final ledger. Admin will only have to fill the database with only employee's details and select their type of work and their post and final report will be prepared by the generation module.

As codes has been developed by using the concepts of object-oriented programming so, coding reusability can easily be achieved. Even if there will be changes to made in future, it can be performed by the technical person within few seconds just by changing the working rules. All working departments within the organization will share the common file to access the particular data, so there is no chance of data redundancy and helping in maintaining data consistency.

Each working section can be done by selecting only options and its graphical user interface will make the system unique as compared to other system.

TABLE OF CONTENT

S.No.	Content	Page No.
I	List of Figures	8
II	List of Tables	9
1	OVERVIEW OF THE PROJECT	10-13
1.1	Objective	11
1.2	Introduction	12
1.3	Problem of The Manual System	13
2	SYSTEM STUDY AND ANALYSIS	14-29
2.1	Introduction	15
2.2	Feasibility Study <ul style="list-style-type: none"> • Technical Feasibility • Operational Feasibility • Economical Feasibility • Legal Feasibility • Schedule Feasibility 	16
2.3	Other Feasibility Factors <ul style="list-style-type: none"> • Market and Real Estate Feasibility • Resource Feasibility • Cultural Feasibility • Financial Feasibility 	18
2.4	Output	19
2.5	Hardware And Software Requirements <ul style="list-style-type: none"> • Hardware Requirements • Software Requirements 	20
2.6	Problem of The Manual System	21
2.7	Solutions of The Problems Occurred in Manual System	22
2.8	The New System <ul style="list-style-type: none"> • Need • Performance • Efficiency • Security 	23
2.9	System Project Selection	24
2.10	Analysis of System <ul style="list-style-type: none"> • System Analysis Phase • Procedures in The Existing System • System Requirement Analysis 	25

	<ul style="list-style-type: none"> • Analysis in This Project • Data collection 	
3	SYSTEM DESIGN	30-56
3.1	Introduction <ul style="list-style-type: none"> • Logical Design • Physical Design • Design/Specification Activities 	31
3.2	Input Design	33
3.3	Output Design	33
3.4	Design Methodology	34
3.5	Design Overview <ul style="list-style-type: none"> • Flowchart • Data Flow Diagrams • ER Diagram 	34
3.6	Interface <ul style="list-style-type: none"> • Output Definition • Input Definition 	42
3.7	Database and File Design <ul style="list-style-type: none"> • Database Design • Program File Design 	52
3.8	Software Description <ul style="list-style-type: none"> • Coding Standards 	53
4	SYSTEM TESTING AND IMPLEMENTATIONS	57-64
4.1	Testing	58
4.2	Types Of Testing <ul style="list-style-type: none"> • Unit Testing • Integration Testing • Validation Testing • Recovery Testing • Security Testing • Stress Testing • Black Box Testing • Test Data Output 	58
4.3	System Implementation Requirement	60
4.4	Maintenance Phase	61
4.5	Conclusion	62
4.6	Scope of Improvement	62
4.7	Scope of Future Application	63
4.8	User Manual	63

	REFERENCES	
	Source Code	

LIST OF FIGURES

Figure No.	Description	Page No.
1.1	Flowchart	35
1.2	Flowchart (Contd.)	36
2	Symbols Used In DFD	38
3	Data Flow Diagram for PMS	39
4	Symbols used in ER Diagram	40
5	ER Diagram for PMS	41

LIST OF TABLES

Table No.	Description	Page No.
1	Table of Content	6
2	List Of Figures	9
3	List of Tables	10
4	Database Design	52

OVERVIEW OF THE PROJECT

1.1 OBJECTIVE

Payroll management system explain itself that it's an accounting package that will provide a financial solutions for a particular organization or for a particular institutional body. This payroll management helps accounts to maintain all records by using predefined rules set by the organization working pattern.

Its admin module has been developed in such a pattern that allows to change the working rules any time as per the business requirement. Its automatic rates and salary are fixed which saves processing time for preparing final ledger. Admin will only have to fill the database with only employee's details and select their type of work and their post and final report will be prepared by the generation module.

As codes has been developed by using the concepts of object oriented programming so, coding reusability can easily be achieved. Even if there will be changes to made in future, it can be performed by the technical person within few seconds just by changing the working rules. All working departments within the organization will share the common file to access the particular data, so there is no chance of data redundancy and helping in maintaining data consistency.

Each working section can be done by selecting only options and its graphical user interface will make the system unique as compared to other system.

1.2 INTRODUCTION

Employee payroll Management is an application that a company maintains in order to give a modern effective management for maintenance of records.

A computerization of company Maintenance is of great significance for making the cumbersome process of employee payroll records maintenance as simple as possible using structured and modular design.

Due to spurring of many books the record maintenance of employee's has grown considerably. Though the whole work cannot be claimed to be project, the main purpose of this work is record maintenance and pay slip generation.

This project helps to change all the manual information into computerization. In "PAYROLL MANAGEMENT SYSTEM," a big number of registers are supposed to maintain. And staff will always rely on the big manual register. But if the system is computerized then staff can easily handle the account of each and every employee working in the company. Maintaining the records for longer period is very difficult, space and time consuming.

Programmer should understand the working of the whole system before starting the project. If the working is not clearly defined, then it may possible that the project produces the result not according to the requirement of the user. So, it is important for creating a successfully and robust project that the problem the user for which the project has to developed should be clearly defined to the programmer. After identifying the need of the potential system, I have developed this project and it tries to accommodate all the need of the existing system.

Company Management receives and maintains data of employee working. It produces payroll slips.

1.3 PROBLEMS OF THE MANUAL SYSTEM

Problems of the manual system are mentioned below:

- i. Preparation of employee details is very difficult and time consuming.
- ii. Preparation of payment slips is very difficult and time consuming.
- iii. Record keeping of employees detail for every month is very difficult
- iv. Lot of paper work has to be done for maintaining the record files.
- v. Storage is also a big problem.
- vi. Security is also a very big problem of the manual system.

SYSTEM STUDY AND ANALYSIS

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Here in the project E-Ticketing, a detailed study of existing system is carried along with all the steps in system analysis. An idea for creating a better project was carried and the next steps were followed.

2.2 FEASIBILITY STUDY

An important outcome of the preliminary investigation is the determination that the system requested is feasible. Feasibility study is carried out to select the best system that meets the performance requirements.

Feasibility study is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. It involves preliminary investigation of the project and examines whether the designed system will be useful to the organization. Months or years of effort, thousand for millions of money and untold professional embarrassment can be averted if an in-conceived system is recognized early in the definition phase.

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations.

Generally, feasibility studies precede technical development and project implementation.

The different types of feasibility are:

- i. Technical feasibility
- ii. Operational feasibility
- iii. Economical feasibility
- iv. Legal feasibility
- v. Schedule feasibility

2.2.1 Technical feasibility

Technical Feasibility deals with the hardware as well as software requirements. Technology is not a constraint to type system development. We have to find out whether the necessary technology, the proposed equipment's have the capacity to hold the data, which is used in the project, should be checked to carry out this technical feasibility.

The technical feasibility issues usually raised during the feasibility stage of investigation includes these.

This software is running in windows 7 Operating System, which can be easily installed.

The hardware required is Pentium based server.

The system can be expanded.

2.2.2 Operational Feasibility

This feasibility test asks if the system will work when it is developed and installed.

Operational feasibility in this project:

- The proposed system offers greater level of user-friendliness.
- The proposed system produces best results and gives high performance.
- It can be implemented easily.

So this project is operationally feasible.

2.2.3 Economical feasibility

Economical Feasibility deals about the economical impact faced by the organization to implement a new system. Financial benefits must equal or exceed the costs. The cost of conducting a full system, including software and hardware cost for the class of application being considered should be evaluated.

Economic Feasibility in this project:

- The cost to conduct a full system investigation is possible.
- There is no additional manpower requirement.
- There is no additional cost involved in maintaining the proposed system.

2.2.4 Legal feasibility

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

2.2.5 Schedule feasibility

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable.

2.3 OTHER FEASIBILITY FACTORS

2.3.1 Market and real estate feasibility

Market feasibility studies typically involve testing geographic locations for a real estate development project, and usually involve parcels of real estate land. Developers often conduct market studies to determine the best location within a jurisdiction, and to test alternative land uses for given parcels. Jurisdictions often require developers to complete feasibility studies before they will approve a permit application for retail, commercial, industrial, manufacturing, housing, office or mixed-use project. Market Feasibility takes into account the importance of the business in the selected area.

2.3.2 Resource feasibility

This involves questions such as how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies,

2.3.3 Cultural feasibility

In this stage, the project's alternatives are evaluated for their impact on the local and general culture. For example, environmental factors need to be considered and these factors are to be well known. Further an enterprise's own culture can clash with the results of the project.

2.3.4 Financial feasibility

In case of a new project, financial viability can be judged on the following parameters:

Total estimated cost of the project

Financing of the project in terms of its capital structure, debt equity ratio and promoter's share of total cost

Existing investment by the promoter in any other business

Projected cash flow and profitability

2.4 OUTPUT

The feasibility study outputs the **feasibility study report**, a report detailing the evaluation criteria, the study findings, and the recommendations.

2.5 HARDWARE AND SOFTWARE REQUIREMENT

2.5.1 Hardware Requirements

Hardware requirement are the basic need of the system or the package, which is been developed and will be deployed upon the system, which should have these basic components or fulfill these basic hardware needs of these package.

The following hardware is recommended for the user.

Processor	Intel Pentium IV 2.4 GHZ or above
Clock speed	500 MHZ
RAM	32MB or above
System bus	32 bits
Monitor	Color monitor
Hard disk	8 GB or above
CD Drive	Any CD ROM
Input device	Key board, Mouse

2.5.2 Software Specification

Operating System	MS WINDOWS XP SP2
Front End	Visual Studio 2012, ASP.Net
Back End	SQL Server
Additional Software	DOTNet framework 3.5

2.6 PROBLEMS OF THE MANUAL SYSTEM

Problems of the manual system are mentioned below:

- vii. Preparation of employee details is very difficult and time consuming.
- viii. Preparation of payment slips is very difficult and time consuming.
- ix. Record keeping of employees detail for every month is very difficult
- x. Lot of paper work has to be done for maintaining the record files.
- xi. Storage is also a big problem.
- xii. Security is also a very big problem of the manual system.

2.7 SOLUTIONS OF THE PROBLEMS OCCURRED IN MANUAL SYSTEM

- i. By the use of computers, preparation of employee details becomes easier and rapidly by making some small programs.
- ii. Payment slips generation can be done easily by computerized payroll management system. This means that these calculations can be done automatically.
- iii. Record keeping becomes very easy and time saving.
- iv. Management information reports are created very easily by giving some instructions to the computer.
- v. There will be no need of paper work for maintaining the record files.
- vi. Storage can be easily done through floppy disk & hard disk having large capacity.
- vii. Important documents can be easily protected.

2.8 THE NEW SYSTEM

2.8.1 Needs

In manual payroll records keeping, there are many paper works to do. The Manager is always engaged in proper maintenance of register. It is very tedious as well as time-consuming work for both the employee and company. For this he/she also requires staff to help and assists him in performing his work.

It is very important for creating a successful and robust project that the problem or need of the user for which the project has to develop should be clearly defined to the programmer. I, being the system analyst, had already that all this hostel needs is one computer which help to get rid them of all junk they are currently using in the company, for systematic storage records.

2.8.2 Performance

A computer works and thinks very much like the human brain and in a much more precise manner. It computes data at the speed of light and can handle millions and millions of bits of information's in one go. If a system can be made to handle records on a computer, it will work at a very fast and can be handle all the different queries, which might be asked by the user. This way, the entire performance of the company can be increase as user sitting on the computer type in queries, and get any information looking for in blind eyes. This saves them the entire hassle of finding one of the hundred file cabinets, then opening it and searching for the record, which they might want. A computer system can increase performance of the company.

2.8.3 Efficiency

When a person using the computer, pressed the enter key, a signal is sent to the processor which is then processed and an output comes. If we could enter data, find data, modify data and a lot of other activities at that kind of speed, it can only make work done much faster and at a very efficient as compared to the cost of the system and the cost of 100 file cabinets. Computer, if applied for the right use, are the most resourceful and proficient machines every created by man. We can get rid of all the file works and all the data is saved into the computer.

2.8.4 Security

Security is also given a great important so that no unauthorized user can tamper the records of the company. This makes, security a big part of the new system. Record can be changed only by the authentication of user who is sitting in front of a system and who is dealing with all the transaction.

2.9 SYSTEM PROJECT SELECTION

Every system project must at some stage have been the object of a selection process. The possible sources for ideas for system projects.

A very common source for ideas for project is request or suggestions from departments within the company. Sometimes a new technical development makes possible a new approach to a problem. A major source of a system projects is the ideas from previous studies.

The criteria for selecting projects are: -

- i. Potential return on investment.
- ii. Management desire.
- iii. Technical feasibility.
- iv. Capacity of the system department to do the project.
- v. Requirement for integration with other systems.
- vi. Critical company need.

Sometimes the user may not be involved actively in this phase as the project may be assigned to him by the organization.

2.10 ANALYSIS OF SYSTEM

2.10.1 System Analysis Phase

Analysis can be referred to as study of a problem and its division into smaller modules so as to solve the particular problem.

A complete understanding of software requirements is necessary to help the new software being developed succeed. No matter how well designed or well coded, a poorly analyzed and specified program will disappoint the user and will also be a burden to the developer.

At this stage comes Requirement Analysis that is a process of discovery, refinement, modeling and specification. Models of the required data, information and control flow are created. It bridges together the gap between system-level software allocation and software design.

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of the phase is the requirement document. Ideally, this document states in a clear and precise fashion what is to build. This analysis represents the “what” phase. The requirement document tries to capture the requirement from the employee’s perspective by defining goals and interactions at a level removed from the implementation details.

The analysis phase is a problem- solving phase. The problem is that in manual payroll record keeping system, excessive staff employment is required, extremely time consuming process is involved, inconveniences to both the employee and the company. So, while analyzing I found many requirements expected from the system like: - to solve the time of both the company and the employee, to perform automatic calculation of the pay, to maintain a proper pay slips of employee record as well as billing of the company. To reduce the whole paper work without the help of the computer is very tedious to maintain and keeping it update.

To minimize any type of inconveniences and to maximize employee oriented services, automatic generation of the patient code and allotted the code to the employee. This will help to maintain to keep the records for the longer period.

The requirement document may be expressed in formal language. Traditionally, the requirement documents are written in English. The requirement document does not specify information at the higher level of description. The problem statement, the employee’s expectation are examples of high- level details. There is a fundamental conflict between high level and low level of details. The requirement document states what the system should accomplish, independent of many of the details.

The discovery process used in establishing the requirements during the analysis phase is best described as a refinement process than as levels of detail process. Things might be expressed as object in an object- based technology where data and algorithms are hidden behind hierarchical- polymorphic methods. In general, the description of things in the system can be much more general and not confined to a particular technology. In a more general sense, this document describes the

ontology that is the noun phrases and the verb phrases that will become the guidelines for defining the application specific protocol.

The requirement descriptions of the things in the system and their actions do not imply an architecture design rather a description of the artifacts of the system and how they behave, from the employee's perspective. Later, in the design phase, these requirement descriptions are mapped into computer science based primitives, such as lists, stacks, trees, graphs, algorithms, and data structures.

The description of the abstraction of the noun phrases and the verb phrases are not bound to the use of the written human languages. Most written human language is too vague to capture the precision necessary to build a system. Alternative description mechanisms based on mathematical logic are sometimes more suitable but much more difficult to accomplish. Mathematical logic provides a scientific foundation for precisely expression information. However, frequently in the real world, a precise description is not attainable. Again the requirements documents should state in a clear and precise fashion what is to build.

The definitive mechanism to author such a documents, either formally or informally, has yet to be developed, although reasonable success has been achieved with existing methods including CASE tools and tools based on mathematical logic.

Later, in the design phase, the very important decomposition of the program leads to the development of the data structures and algorithms. A functional decomposition for a distributed environments leads to a natural split of the data structures and algorithms. An object – based decomposition leads to a natural joining of data structures and algorithms forming objects with methods. The requirement documents should be independent of the decomposition technique. The analysis team develops the requirement documents, which talk about things and action on things. These documents should also included states events typical scenarios of usage.

2.10.2 Procedures In The Existing System

In the “PAYROLL” project, there are 3 classes such as

- i. LINES
- ii. MENU
- iii. EMPLOYEE.

2.10.3 System Requirement Analysis Is Divided Into The Following Areas

i. Initial Investigation (Problem recognition)

The first step for the development of system is the need of a new system. This might be a user's request to change, improve or enhance an existing system. This step determines whether the request is valid and feasible or not before improving or modifying the current system or build a new one.

ii. Determining the User's Information Requirements (Evaluation & Synthesis).

At this stage, the user's requirements are determined with the help of Prototyping, asking questions and obtaining information from the existing System.

- Problem Definition & Project Initiation.

Once the problem is clearly determined, the problem is defined in terms as to what is required, how etc.

- Background Analysis.

Once the project is initiated, the analyst begins to learn about the setting, the existing system and the physical processes related to the revised system. All the background information is collected and thoroughly studied.

After obtaining background knowledge, the analyst begins to collect data on the existing system's outputs, inputs and costs with the help of written documents, on-site observations, interviews and questionnaires.

- Fact Analysis.

After gathering data, they must be organized and evaluated and conclusions drawn for preparing a report to the user for final review and approval. At this stage, some tools are used i.e. Data flow diagrams, Decision tables, Structure chart etc. The end result obtained determines whether the new system is feasible or not.

2.10.4 In This Project, We Have Done The Following Analysis

i. Initial Investigation

We saw that there were very fewer sites available that could give complete information about transacting on net to users. Also there was a need to automate the entire system.

ii. Determination of User's Requirements

We contacted a no. of people about this project who agreed with the idea that this system will help a lot many people rather the entire nation.

iii. Problem definition & Project Initiation

Problem definition is an initial investigation, which defines the problem that led to the user request. It states the objectives the user is trying to achieve and the result user wants to see. It gives more stress on logical requirements than the physical requirements.

The main objective of "PAYROLL MANAGEMENT SYSTEM" is to organize and maintain the records of account holders efficiently. The records are automatically updated in the master file when any transaction takes place.

i. Background Analysis

All the existing details were collected from the respective experts who gave us their brochures, pamphlets, discussions with concerned personnel, details of exactly how information flows from one end- to-the-other. We also searched various kinds of shares and related sites, which did help us but not a very large-scale.

ii. Fact Analysis

All the data and information collected was organized to show how information flows and at what stage, exactly what happens? For this we tried to explain this with the help of Data Flow Diagrams (explained further).

2.10.5 Data Collection

i. OBSERVATION

The software has been prepared with a theory of turning a manual payroll management system into a fully computerized one. The theory behind preparing the project is to make the employee of the company easy to handle the records and services effectively. I have developed this software after keeping in view the need and requirement of large companies. Systems require large number data to be handling. The advantage of the computerized system over manual system is better performance, accuracy, consistency etc.

ii. QUESTIONNAIRES

The manual procedures of the PAYROLL were observed. This was essential to analyze the flow of the information and the actual usages of the software programmers. The actual procedures adopted by the company staff were reviewed and the various registers they maintain were studied. This was helpful to design the reporting requirement from the software.

For the proposed of designing an effective software package, is questionnaire was prepared and different staff of a hospital was interviewed to access their requirement and how the proposed software can benefit them. For example:-

Different people had different answers to the above question. However most people were in favors of computerization of the existing system. For the convenience of the employee working in the company, it must that both hardware and software should work efficiently to provide right information whenever required.

SYSTEM DESIGN

3.1 INTRODUCTION

System Design is the most creative and challenging phase in the system life cycle. Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. System design is a solution *how to approach* the creation of a new system. System design transforms a logic representation of what is required to do into the physical specification. The specification is converted into physical reality during development.

3.1.1 Logical Design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- i. Reviews the current physical system – its data flows, file content, volumes, frequencies etc.
- ii. Prepares output specifications – that is, determines the format, content and frequency of reports.
- iii. Prepares input specifications – format, content and most of the input functions.
- iv. Prepares edit, security and control specifications.
- v. Specifies the implementation plan.
- vi. Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- vii. Reviews benefits, costs, target dates and system constraints.

3.1.2 Physical Design

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- i. Design the physical system.
- ii. Specify input and output media.
- iii. Design the database and specify backup procedures.
- iv. Design physical information flow through the system and a physical design Walk through.
- v. Plan system implementation.
- vi. Prepare a conversion schedule and target date.
- vii. Determine training procedures, courses and timetable.
- viii. Devise a test and implementation plan and specify any new hardware/software.
- ix. Update benefits, costs, conversion date and system constraints.

3.1.3 Design/Specifications Activities

- i. Concept formulation.
- ii. Problem understanding.
- iii. High level requirements proposals.
- iv. Feasibility study.
- v. Requirements engineering.
- vi. Architectural design.

3.2 INPUT DESIGN

Input Design deals with what data should be given as input, how the data should be arranged or code, the dialog to guide the operating personnel in providing input, methods for preparing input validations and steps to follow when error occur. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system. It is achieved by creating user friendly GUI (Graphical User Interface). The goal of the designing is to make the interface as simple and attractive as possible. The user can browse through the items available on the website, add or remove them to and from the cart as required. It provides invoices also.

When the user click on the Checkout button the session checks for a valid login. If the user has logged in it goes to the payment gateway else it redirects to the login page. After a successful login the payment gateway launches.

3.3 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. The objective of output design is to convey information about past activities, current status or projections of the future, signal important events, opportunities, problems, or warnings, trigger an action, confirm an action etc. Efficient, intelligible output design should improve the system's relationship with the user and helps in decisions making. In output design the emphasis is on displaying the contents of the website in the best possible manner. The primary consideration in design of output is the information requirement and product descriptions for the end users. The major formation of the output is to convey the information and so its layout and design need a careful consideration.

3.4 DESIGN METHODOLOGY

Database and database system have become essential component of everyday life in modern society. In the courses of a day, most of us encountered several activities that involved some interaction with the database. There are several ways of implement database. Some of them are file handling mechanism, relational database, object relational database. In our project we will use file-handling features provided by C++ languages. The project is developed by in such a way to provide a very efficient user interface of the project.

In this project there are 3 classes

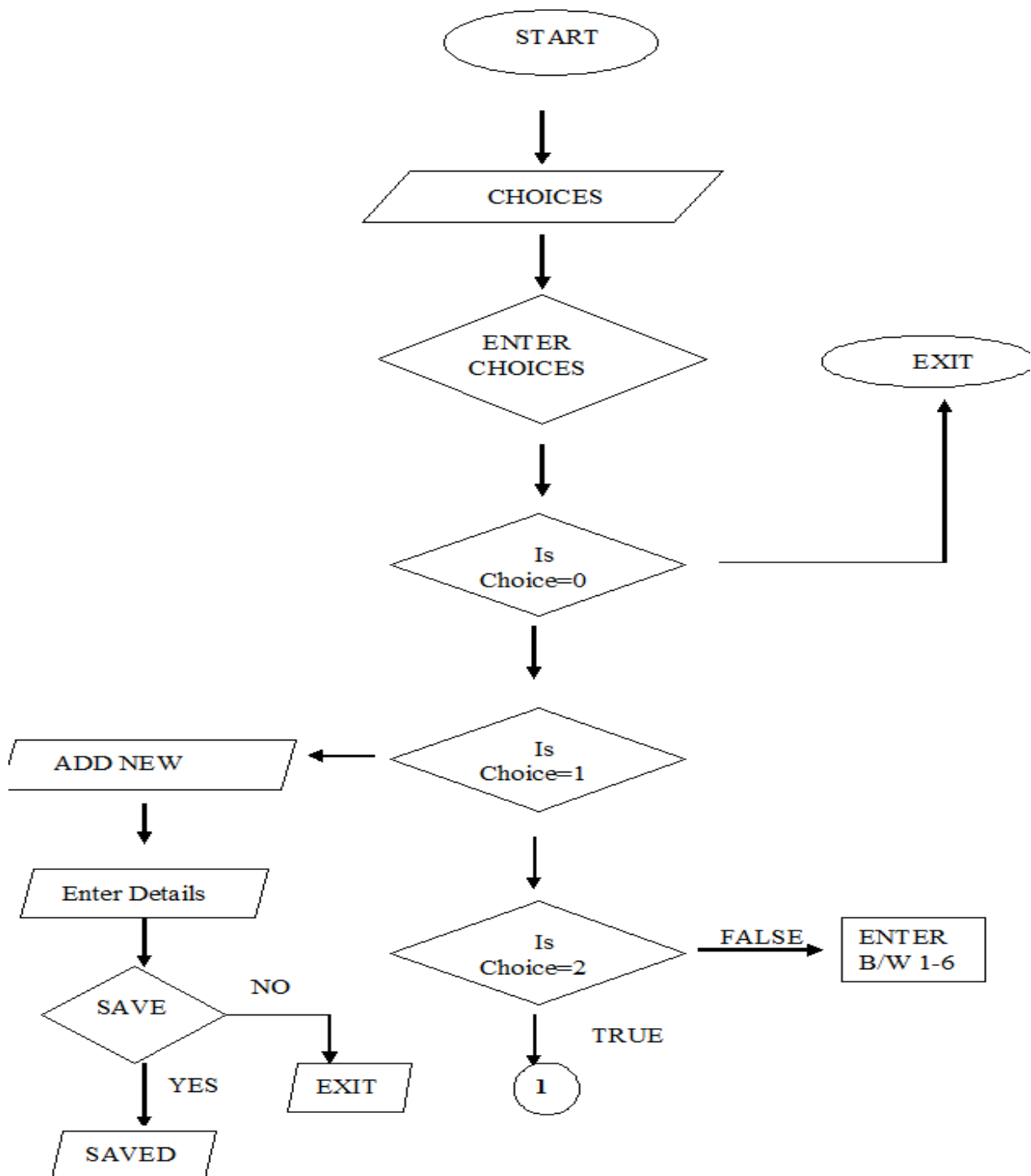
- i. Lines class = this class contains drawing functions.
- ii. Menu class = It performs function relating to different menu in the project. In this employee, this class controls designation & salary classes as all the display functions.
- iii. Employee class = It performs function relating employees.

3.5 DESIGN OVERVIEW

- i. Flow Chart
- ii. DFD
- iii. E-R Diagram

3.5.1 Flowchart

In procedural languages program, is started with the first line and follow a predefined path. Flowchart is used to define that predefined path and it show the flow of control throughout the program.



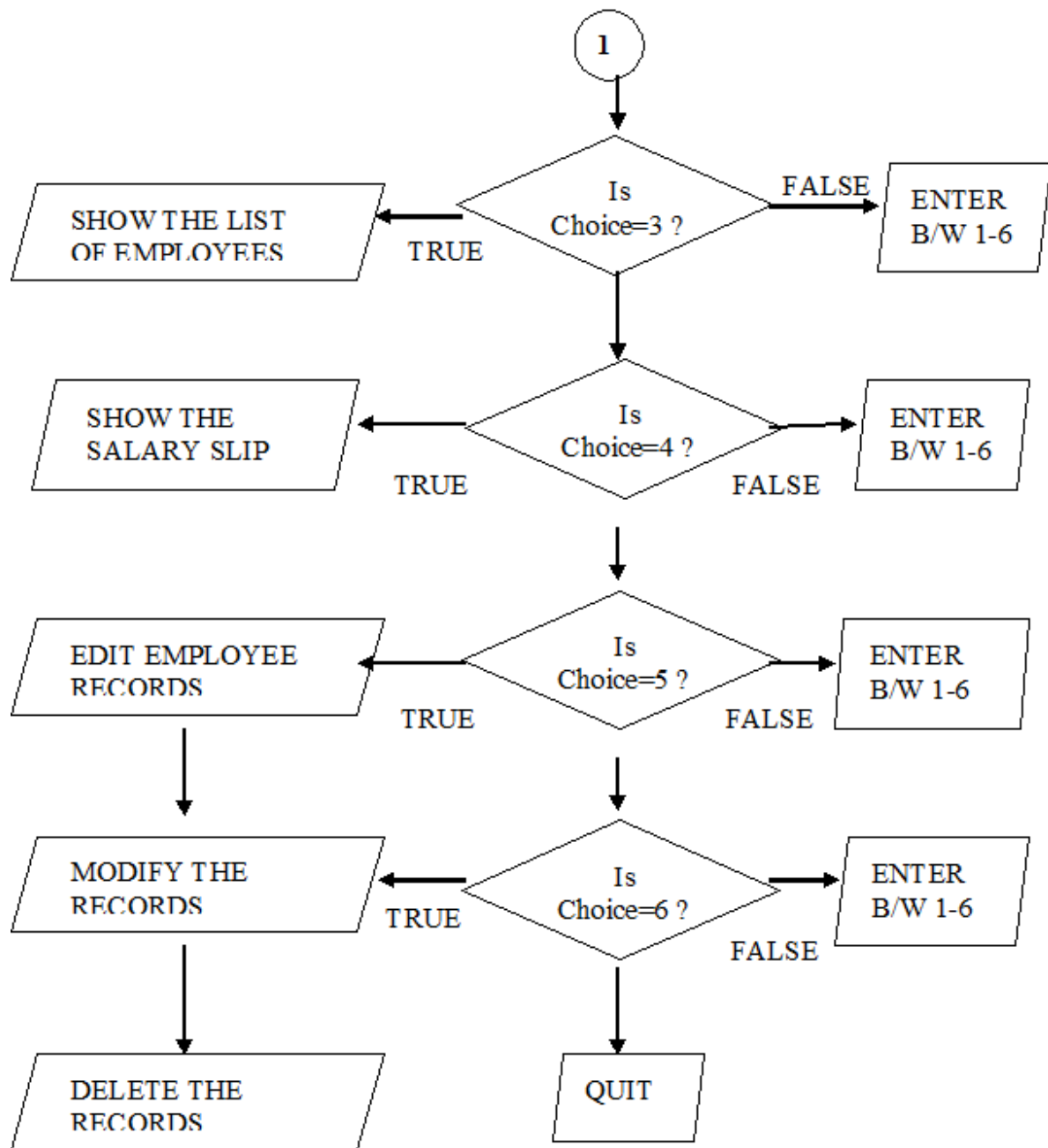


Fig 3.1 FLOWCHART FOR PAYROLL MANAGEMENT SYSTEM

3.5.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

Data Flow Diagram serves two purposes:

- i. To provide annunciation of how data are transformed as they move through the system.
- ii. To depict the functions that transforms the data flow.

DFDs are an excellent mechanism for communicating with the customer during requirement analysis and are widely used for the representation of external and top-level internal design specification. In the latter situations, DFDs are quite valuable for subsystem, files and data links. The DFD methodology is quite effective, especially when the required design is unclear. In the process, many levels of DFDs are created depending upon the level of details needed

The Level 0 DFD is also called Context Level DFD. It depicts the overview of the entire system. The major external entities, a single process and the output stores constitute the level-0 DFD. Though this diagram does not depict the system in detail, it represents the overall inputs, process and output of the entire system at a very high level.

The Level 0 DFD is now expended into a level 1 model. It should be noted that information flow continuity is maintained between level 0 and level 1. The process represents at DFD level 1 further refined into lower levels. This further refinement is continued until an easily implement able program component is reached.

3.5.2.1 Symbols Used In DFD

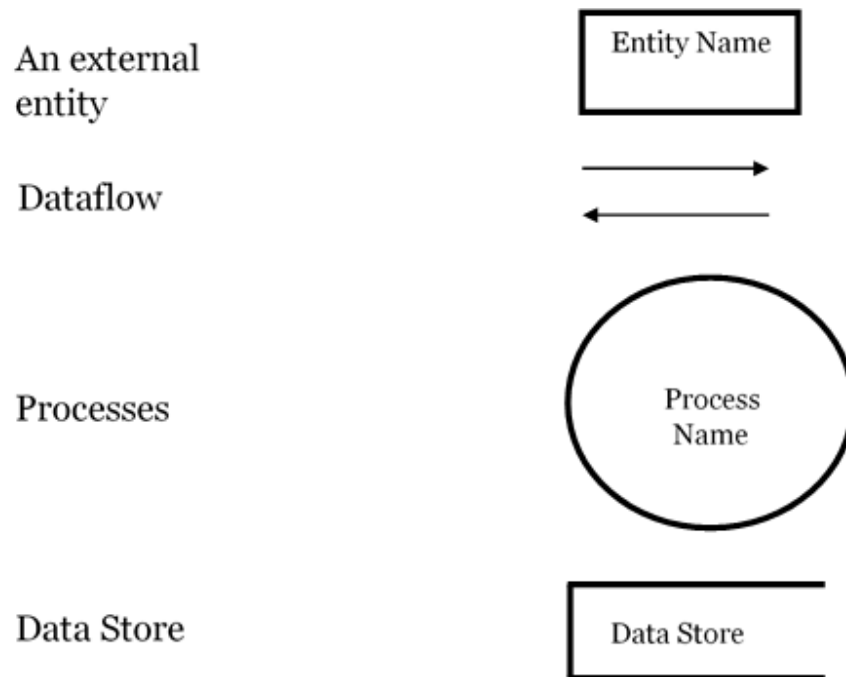


Fig 3.2 SYMBOLS USED IN DFD

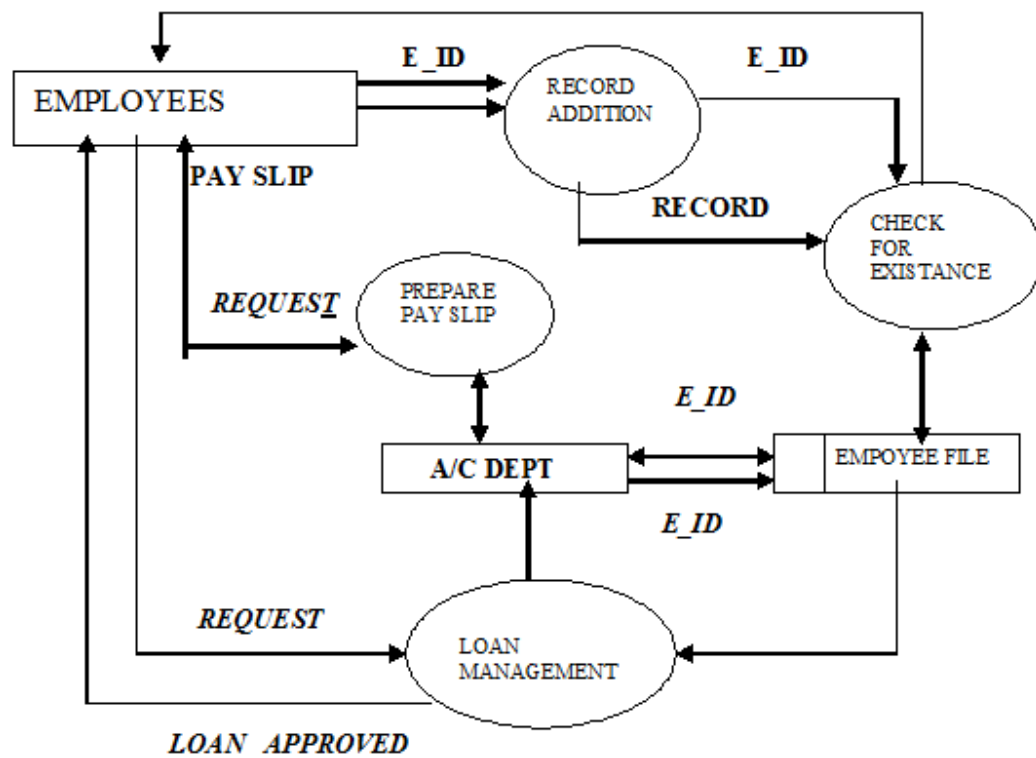


Fig3.3 DATA FLOW DIAGRAM FOR PAYROLL MANAGEMENT SYSTEM

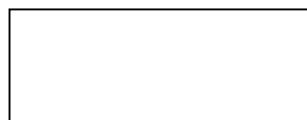
3.5.3 Entity Relationship Diagrams (Er-Diagrams)

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes

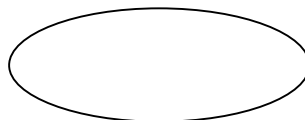
An **entity-relationship model** (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

Symbols used in this E-R Diagram

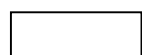
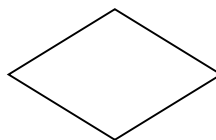
Entity: Entity is a “thing” in the real world with an independent existence. An entity may be an object with a physical existence such as person, car or employee. Entity symbol is as follows



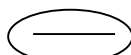
Attribute: Attribute is a particular property that describes the entity. Attribute symbol is



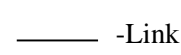
Relationship: Relationship will be several implicit relationships among various entity types whenever an attribute of one entity refers to another entity type some relationship exists. Relationship symbol is:



-Table



Primary Key



-Link



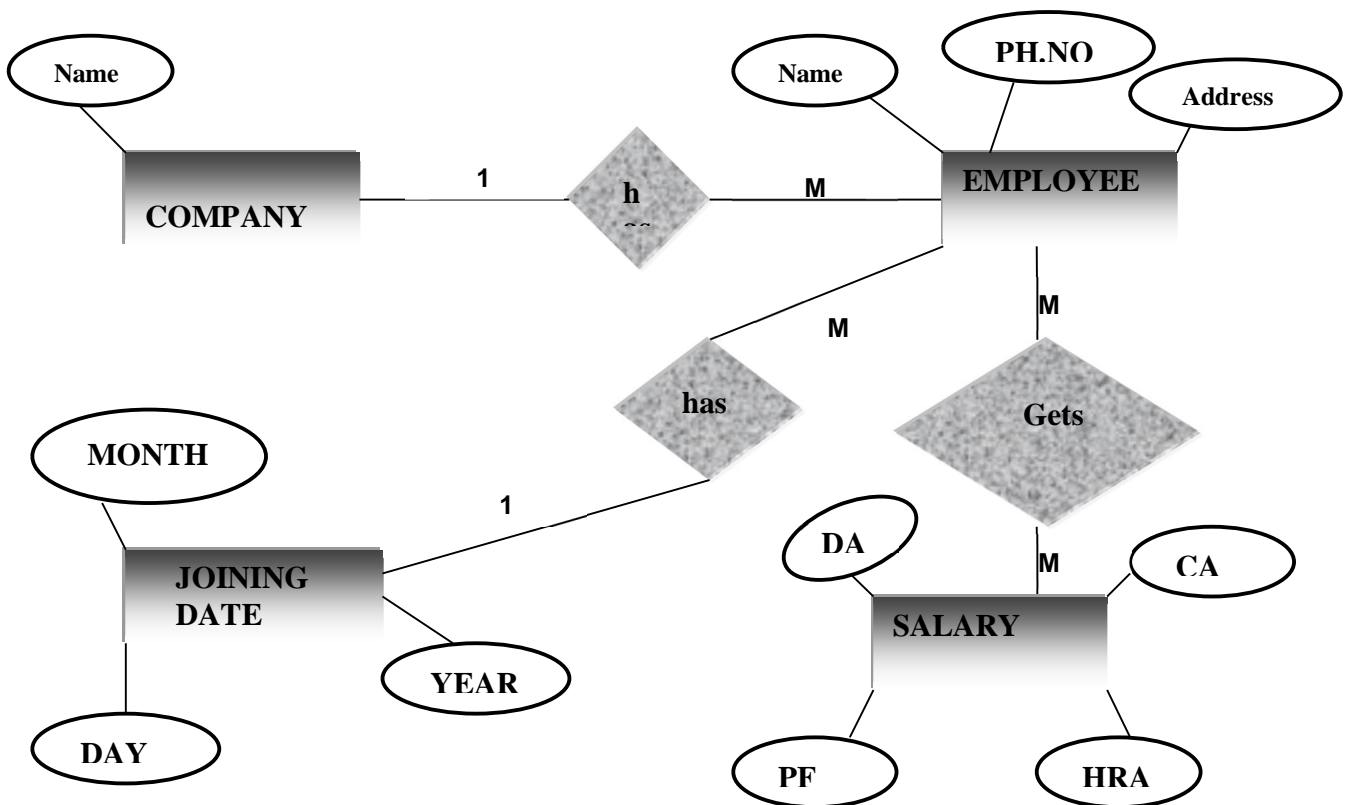
Fields



-Relationship

M: N-Cardinality Relation

Fig 3.3 ER DIAGRAM FOR PAYROLL MANAGEMENT SYSTEM



3.6 INTERFACE

3.6.1 Output Definition

In output definition following screen are shown

- i. Main Menu Screen.
- ii. New employee Record Screen.
- iii. Display employee Record Screen.
- iv. List of employee Screen.
- v. Salary slip screen.
- vi. Edit screen.
- vii. Quit screen.

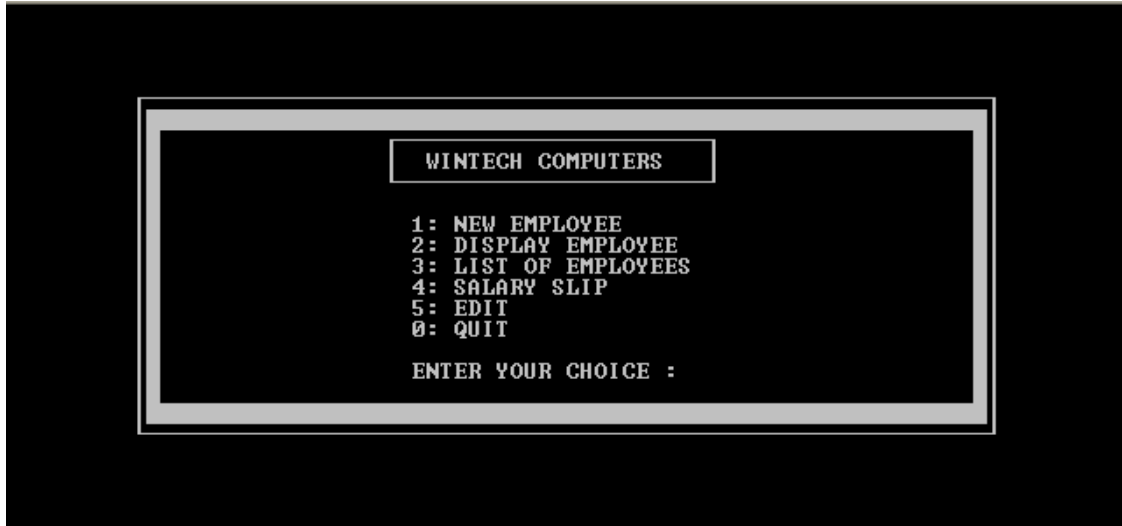
3.6.2 Input Definition

In input definition following screen are shown

- i. Main Menu Screen.
- ii. New employee Record Screen.
- iii. Display employee Record Screen.
- iv. List of employee Screen.
- v. Salary slip screen.
- vi. Edit screen.
- vii. Quit screen.

- i. MAIN SCREEN

Just after run the software user will see the main screen. In the main screen, all the options available in this project will be available. User will use the keys to move between the choices & press enter to select a particular choice.



- Enter 1 to add new records.
- Enter 2 to display the records of any particular employee.
- Enter 3 to display the list of all the employee working in the company.
- Enter 4 to display the salary slip of any particular employee.
- Enter 5 to edit the record of any particular employee.
- Enter 0 to quit from the program.

ii. NEW EMPLOYEE RECORD SCREEN

If the user input 1 for the first option in the main screen then he will see the new employee record screen. In this screen, the required detail of the new employee is entered. At the bottom of the screen, message is shown to guide the user.

```

                                ADDITION OF NEW EMPLOYEE                                <0>=EXIT
Employee Code # 3
~~~~~
Name      : Rankey
Address   : n-82,sth ext.,new delhi
Phone no. : 25331123

JOINING DATE
~~~~~
Day       : 12
Month     : 11
Year      : 2007

Designation : manager
Grade       : a
House (y/n) : y
Convence (y/n) : y
Loan        : 50000
Basic Salary : 10000

Do you want to save (y/n) _
```

iii. DISPLAY EMPLOYEE RECORD SCREEN:

If the user input 2 enter at the second option in the main screen then he will see the details of the employee working in company .in this section program want from the user to enter the ID or CODE of the employee whose record he want to see., the program will show another screen. In that screen the entire details of that employees will be show. 'Press any key to continue' and if user presses any key than the program will jump to main screen.

```

                                                                    <0>=EXIT
Enter code of the Employee 3_
```

Now press Enter key to continue.....and user will see this screen where all the details related to the employee will be there.

```

Employee Code # 3
~~~~~
Name       : RANKEY
Address    : N-82,STH EXT.,NEW DELHI
Phone no.  : 25331123

JOINING DATE
~~~~~
Day   : 12
Month : 11
Year  : 2007

Designation : MANAGER
Grade       : A
House (y/n) : Y
Converse (y/n) : Y
Loan        : 50000
Basic Salary : 10000

Press any key to continue...
```

iv. LIST OF EMPLOYEE RECORD SCREEN:

If the user input 3 for the third option in the main screen then he will see the list of all the employees working in the company.

LIST OF EMPLOYEES						
CODE	NAME	PHONE	DOJ	DESIGNATION	GRADE	SALARY
1	SUMIT	25334413	12/9/2007	MANAGER	A	20000
2	ANKIT BHALLA	25334413	1/11/2007	MANAGER	B	20000
3	RANKEY	25331123	12/11/2007	MANAGER	A	10000

Press any key to continue...

Now press Enter key to continue.....

v. SALARY SLIP SCREEN:

If the user input 4 for the fourth option in the main screen then the user will see the salary details of any particular employee. In this screen, the program want to enter the employee ID or CODE to see all the details relating to that employee, than the program will show the detail of that employee.

```

                                <0>=EXIT

Enter code of the Employee  2
  
```

Now press enter key to continue with this.....and user get another containing all the details of employees related to their salary details

WINTech COMPUTERS	
SALARY SLIP December, 2007	
Date: 29/11/2007	
Employee Name : ANKIT BHALLA	
Designation : MANAGER	Grade : B
<div style="border: 1px solid black; padding: 5px;"> <p>Basic Salary : Rs.20000</p> <p>ALLOWANCE</p> <p> HRA : Rs.0</p> <p> CA : Rs.0</p> <p> DA : Rs.1000 Rs.1000</p> <p>DEDUCTIONS</p> <p> LD : Rs.300</p> <p> PF : Rs.400 Rs.700</p> <hr/> <p>NET SALARY : Rs.20300</p> </div>	
CASHIER	EMPLOYEE

Now press Enter key to continue.....

vi. EDIT SCREEN:

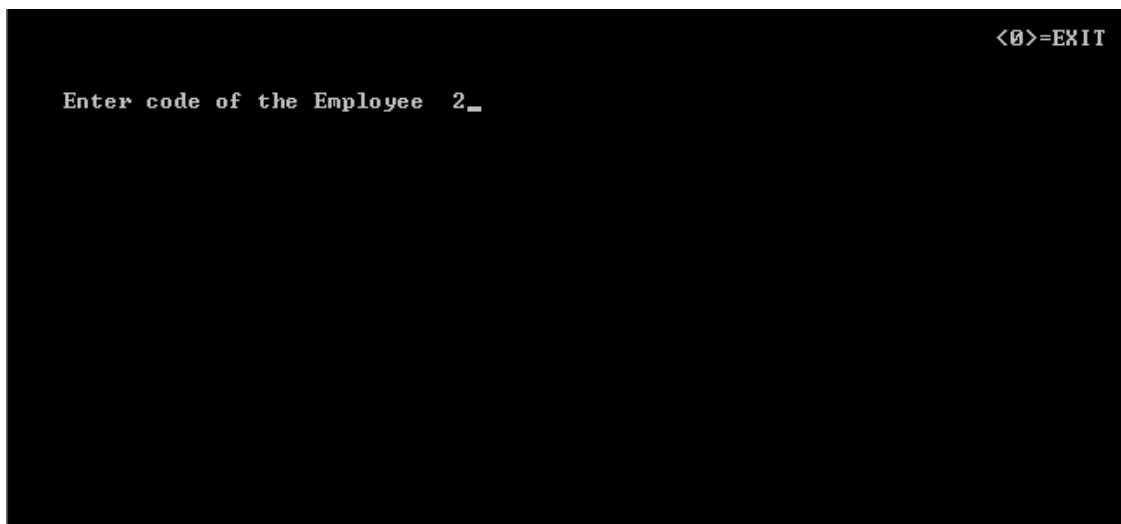
If the user input 5 for the fifth option in the main screen then the user will see the edit sub screen. This menu will have 3 option:-

- Delete record
- Modify record
- Exit.

If user input 1 to delete the record of any employee...



This screen will display..... then the user will see that the program is asking to the user that which record of the employee he wants to delete and enter the ID or CODE of that employee.



And now the program is asking the user that he is sure that he or she want to delete the record of that employee, if user enter Y for YES that the record will be deleted from the database or the user enter N for NO, than program will not delete the record of that employee.

```

                                DELETION OF THE EMPLOYEE RECORD                                <0>=EXIT

Employee Code # 2
~~~~~
Name       : ANKIT BHALLA
Address    : NEW DELHI
Phone no.  : 25334413

JOINING DATE
~~~~~
Day   : 1
Month : 11
Year  : 2007

Designation : MANAGER
Grade       : B
House (y/n) : N
Converse (y/n) : N
Loan       : 2000
Basic Salary : 20000

Do you want to delete this record (y/n)

```

And this screen is displaying that the record is deleted

```

Record Deleted
Press any key to continue...

```

Now press any key to continue.....

If user input 2 to modify the record of any employee...

This screen will display..... then the user will see that the program is asking to the user that which record of the employee he wants to modify and enter the ID or CODE of that employee.

```

                                                                    <0>=EXIT

Enter code of the Employee 1_
```

And now the program is asking the user that he is sure that he or she want to delete the record of that employee, if user enter Y for YES that the record will be deleted from the database or the user enter N for NO, then program will not delete the record of that employee.

```

                                                                    <0>=EXIT

                MODIFICATION OF THE EMPLOYEE RECORD

Employee Code # 1
~~~~~
Name       : SUMIT
Address    : DELHI
Phone no.  : 25334413

JOINING DATE
~~~~~
Day   : 12
Month : 9
Year  : 2007

Designation : MANAGER
Grade       : A
House (y/n) : N
Converse (y/n) : N
Loan        : 0
Basic Salary : 20000

Do you want to modify this record (y/n) _
```

If user press yes to modify the record than this screen will display.
In this screen if user wants to change any of the record than he or she can changes it or if no then simply press enter key to continue.....

```
Employee Code # 1          JOINING DATE : 12/9/2007
~~~~~                     ~~~~~
Name      : -
Address   :
Phone no. :
Designation :
Grade     :

Loan      :

Enter the name of the Employee or <ENTER> FOR NO CHANGE
```

And at last program will ask to the user that he or she want to save the record then press YES or just simply press NO to continue

vii. QUIT SCREEN:

If user input 6 for the sixth option in the main menu screen then the user will quit or exit from the software or from the program.

3.7 DATABASE AND FILE DESIGN

3.7.1 Database Design

EMPLOYEE.dat = This file will store the information related to employee.

EMPLOYEE DETAILS	DESCRIPTION
NAME	Name of the employee
ADDRESS	Address of the employee
Phone number	Phone number of the employee
Joining date	Joining date of the employee
Designation	Designation of the employee
Grade	Grade of the employee
Loan	Loan issued for the employee

3.7.2 PROGRAM FILE DESIGN

The programming language used for developing "Payroll management system" software is C++. The programmer in developing in the project is using a very fine and step-by-step Methodology. While making this project it was kept in mind, that user should not find any difficulty while using this software. The project is much simple and well-formed option to use the project.

3.8 SOFTWARE DESCRIPTION

3.8.1 Coding Standards

We have used C++ to develop this program. C++ is object oriented language, so we are discussing some of the concept of object oriented design.

i. OBJECT AND CLASSES

WHAT IS AN OBJECT?

An object is something that has a fix shape & well define boundary. If you look at your computer desk, you would notice several objects of varying descriptions. They may be connected to, or related to, adjacent objects. For example, a keyboard may be attached to a PC, and it is also easy to see different parts of PC are distant objects. You will observe too characteristic things about of objects. Firstly, each object has certain distinctions that enable you to recognize and classified it.

In computer terms, an object is a collection of related variable and function bound together to form a higher-level entity. The variable defines the state of the object, while the function defines the action that can be performed on the object. C++ programming language involved a building a program or a system from object, in a same way that we might build a car from, its components. Thus we can think of a program as being constructed from a set of interrelated part, which activated each other by sending messages to one another. Hence when a program in C++ is running, its work with components sending code to other components and causing them to execute. It can also allocate a new object. This is achieved by using constructor and destructors; this determined how the object of a class are created, copied and destroyed. Thus, object oriented programming is a technique for writing 'good' programs for a set of problem.

ii. WHAT IS A CLASS?

The concept of class is best understood with an analogy. Out of the several objects in a room, let us talk about the picture on the wall. There is a class, which can call the class of "picture" of which the picture on the wall is an instance. The picture in the room belongs to the class of picture, which contains all pictures everywhere. Thus in C++ it is said, "An object is an instance of class", which mean in the example that "a picture is an occurrence of the ideal picture".

iii. OBJECTS DECLARED AS A CLASS

In C++ programming, a class is a definition from which object can be created. An example would be "customer" class consisting of the definition of the customer data and the process, which may act on it into a single unit. An individual object would have values defied for Customer Identification (ID), Name, and Address. When the customer class receives a message, it would activate the corresponding method ton do this action. We shall try to understand, by an example in C++, as to how the idea explained above is used. We have a function that gives us the square root of a value, and we might declare it as:

```
Float square root ();
```

The above function works out square roots. It is recognized by the name "Square root" and returns a value in the form of a floating-point number. We can use it in the following assignment statement:

```
x=square root(y);
```

The above statement in C++ has the effect of calling the function “square root” and passing it the value of the value of the variable “y”. the function works out the square root of that value and passes it back as a floating point number. That value is then assigned to “x”. The point to note is: (a) the function can be used exactly as shown above, i.e., the function is used directly within the language as if it were a variable. (b) All function within C++ has a data type, which determines the form of the return value. For example, in the above case of square root it is of the floating-point type.

An object when executed returns a value. The class defines the type of data returned by an object. A class can be for any data type. The data type of a function in C++ is similar to the class of an object, because the class of an object defines the data type.

iv. DIFFERENCES BETWEEN OBJECT AND FUNCTIONS

The main difference between objects and functions is that a function is usually written to perform only a single service such as the calculation of a square root. A class offers a range of different services, each of which is activated by a different message. The function that an object carries out is thus determined by the message that is sent to it. As an example, the class “number” could offer addition, subtraction, etc as services. Each would be invoked by a different message which would include a number as a parameter. Thus “number +1” would have the meaning: add 1 to the current value of number. “+1” is the message which is sent to the object “number”, the “+” invokes the service of addition. Note that “1” is also an object which has value “1”; it is an instance of the class “number”. Thus for each class all the services that can be provided are defined once and for all. They are coded into the class and are never coded elsewhere. This is applied to any service, which you may wish to provide on any organized data, no matter how complex it is. It could be printing or reading a record. It may also be searching a value in a table in a particular order. The two other parts defined in a class are:

- Messages
- Methods

v. OBJECTS VS CLASSES:

You probably noticed that the illustrations of objects and classes look very similar. And indeed, the difference between classes and objects is often the source of some confusion. In the real world, it’s obvious that classes are not themselves the objects they describe. A blueprint of the bicycle is not a bicycle. However, it’s a little more difficult to differentiate classes and objects in software. This is partially because software objects are merely electronic models of real-world objects in the first place. But it’s also because the term “object” is sometimes used to refer to both classes and instances.

vi. TWO APPROACHES TO SOLVE A PROBLEM WITH PROGRAMMING:

- Structured (Procedural) Methodology – C, Pascal, FORTRAN, and C++.
- Object-oriented Methodology – C++, Smalltalk, JAVA.

vii. PROGRAMMING STYLES:

Here we have listed some general guidelines that should be followed while coding.

- NAMES:

Variables names should be closely related to the entity they represent and module names should reflect their activities.

- CONTROL CONSTRUCTS:

It is desirable that as much as possible single-entry-exit constructs to be used. It is also desirable to use few standard controls construct rather than using a wide variety of constructs just because they are available in the language.

- GOTO'S:

Go to should be used sparingly and in discipline manner. Only when the alternative of using goes to is more complex should the go to be used.

- USER DEFINED TYPES:

Most programming language allows user to define types like the enumerated type. When such facilities are available, they should be exploited where applicable. For example when working with dates, type can be defined for the day of the week in Pascal, this can be done as follows:

Type month = (Jan, Feb, Mar Apr, May, June, July).

- NESTING

The different control constructs, particularly the If-Then-Else, can be nested. If the nesting becomes too deep, the program becomes harder to understand. Where possible deep nesting should be avoided, even if it means a little inefficiency.

- ROBUSTNESS

A program is robust if it does something planned even for exceptional conditions. A program should check for validity of inputs, where possible, and should check for possible overflow of the data structures. If such situation do arise the program should not “crash”, it should produce meaningful message and exit gracefully.

- INTERNAL DOCUMENTATION

In the coding phase the output is code itself. However, some amount of internal documentation in the code can be extremely using full in enhancing the understandability of the programs. Internal documentation of the program is done by the use of comments. The purpose of comments is to explain what the code is doing, not how it is doing it.

- ADDTION STANDARDS USED IN THE SYSTEM DEVELOPMENT:-

- a. TABLE NAME IS SELF EXPLANATORY

This line conveys that the table name used at the design time, itself tells one what are the contents.

- b. FUNCTION NAME IS SELF EXPLANATORY

The line signifies that the function used in the coding to perform a particular task, can be identified by its name. The name itself specifies about its operation to be performed.

For e.g. Display () -> the name display tells about its operation i.e. display of the outcome on the user's screen.

- c. VARIABLE NAME IS SELF EXPLANATORY

This line signifies that the variable name used at the coding time, itself tells one what type of value it may hold.

SYSTEM TESTING AND IMPLEMEN- TATIONS

4.1 TESTING

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts. Two classes of inputs are provided to test the process:

- i. A software configuration that includes a software requirement specification, a design specification and source code.
- ii. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

4.2 TYPES OF TESTING

Testing is divided into several distinct operations:

4.2.1 Unit Testing

Unit test comprises of a set tests performed by an individual program prior to the integration of the unit into large system. A program unit is usually the smallest free functioning part of the whole system. Module unit testing should be as exhaustive as possible to ensure that each representation handled by each module has been tested. All the units that makeup the system must be tested independently to ensure that they work as required.

4.2.2 Integration Testing

Integration testing is a system technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole. Bottom-up integration consists of unit test followed by testing of the entire system. A sub-system consists of several modules that communicated with other defined interface.

4.2.3 Validation Testing

After validation testing, software is completely assembled as a package, interfacing errors that have been uncovered and corrected and the final series of software test; the validation test begins. Steps taken during software design and testing can greatly improve the probability of successful integration in the larger system. System testing is actually a series of different tests whose primary purpose is to fully exercise the compute –based system.

4.2.4 Recovery Testing

It is a system that forces the software to fail in a variety of ways and verifies that the recovery is properly performed.

4.2.5 Security Testing

It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The system's security must of course be tested from in vulnerability form frontal attack.

4.2.6 Stress Testing

Stress tools are designed to confront programs with abnormal situations. Stress testing executes a system in a manner that demands resources in abnormal quantity and volume.

4.2.7 Black Box Testing

Black box testing is done to find out the following information as shown in below:

- Incorrect or missing functions.
- Interface errors.
- Errors or database access.
- Performance error.
- Termination error.

The mentioned testing is carried out successfully for this application according to the user's requirement specification.

4.2.8 Test Data Output

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

4.3 SYSTEM IMPLEMENTATION REQUIREMENT

The implementation phase is less creative than system design. It is primarily concerned with user training, site preparation, and file conversion. When the candidate system is linked to terminals or remote sites, telecommunication network and test of the network along with the system are also included under implementation.

During the final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place about the same time the user is being trained or late.

In the extreme, the programmer is viewed as someone who ought to be isolated from other aspects of the system development. Programming is itself design work, however. The initial parameters of the candidate system should be modified as a result of programming efforts. Programming provides a 'reality test' for the assumptions made by the analyst. It is therefore a mistake to exclude programmers from the initial system design.

The implementation phase is less creative than system design. It is primarily concerned with user training, site preparation, and file conversion. When the candidate system is linked to terminals or remote sites, telecommunication network and test of the network along with the system are also included under implementation.

During the final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place about the same time the user is being trained or late.

In the extreme, the programmer is viewed as someone who ought to be isolated from other aspects of the system development. Programming is itself design work, however. The initial parameters of the candidate system should be modified as a result of programming efforts. Programming provides a 'reality test' for the assumptions made by the analyst. It is therefore a mistake to exclude programmers from the initial system design.

As we know, creating software is one thing and the implementation of the created software is another. The process of implementation software is much difficult as compared to the task of creating the project.

First we have to implement the software on a small scale for removing the bugs and other error in the project and after removing them we can implement the software on a large scale. We have to implement our project named "Payroll Management System" on a small scale now, to see whether it is working fine or not without any errors, and after testing and evaluating the software we will implement it on a large scale.

Whenever we develop software or project certain hardware and the programmer for developing the software is using software. The hardware and software to be used by the programmer for developing the software should be such that it would result in the development of a project, which would satisfy all the basic needs for which the programmer has created the project. The hardware should be such that cost constraints of the employee should also be taken into account without affecting the performance.

For the development of the software program, we have selected C++ because C++ has a wider acceptance in the Industry. Operators cannot make changes in the Database structure, and hence a great deal of security is maintained. Everything has to be operated through the software itself. Also EXE.file can be created and the same can be delivered to the user.

No additional software is required for this Package. The user need have only the basic operating software on his computer. The developed software runs under DOS as well as WINDOW OS. Whenever we develop software or project certain hardware and the programmer for developing the project is using software. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the programmer has created the project.

4.4 MAINTENANCE PHASE

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. Like any system, there is an aging process that requires periodic maintenance of hardware and software. If new information is inconsistent with the design specification, then changes have to be made. Hardware also required periodic maintenance to keep in tune with design specification. The importance of maintenance is to continue to bring the new system to standers.

User priorities, changes in organization requirements, or environmental factors also call for the system enhancements (To contrast maintenance with enhancement).

4.5 CONCLUSION

After developing the software it is very essential to evaluate it to know whether it solves the defined problem for which it has been created or not. By evaluating the software we can also know the extent to which the created software solves the given problem. The software named “Payroll Management System” has also been evaluated by the programmer in the following ways:-

The project has been compiled and being run on the same hardware platform to be used at the client site by the programmer. This is very essential to do as the programming language i.e. C++, in which the software has been created is not platform independent.

The project has also been compiled and run in the presence of the client for whom the project has been prepared. The Time period used for evaluating the software has been 3 weeks. Thus, after evaluating the project I could conclude that my project is feasible & justified.

4.6 SCOPE OF IMPROVEMENT

There are certain limitations about my project & they are as follows:-

- i. The project do not follow all the aspect of oop:-
There is always a room for improvement of things to implement the new things. So, this project is made in C++ but do not follow all the concepts of OOP or any other technical approach such as Inheritance, Polymorphism.
- ii. The project needs more GUI interface
Since mouse can not be used in this current system. So, it becomes little bit difficult to perform data entry without clickable. This is due to time limitations can be improved in future.
- iii. The project needs a database backhand:-
In our project we have used C++; it uses the file handling features of C++. It is a very obsolete technology for database management. But it doesn't support multi user environment.

4.7 SCOPE OF FUTURE APPLICATION

At the end of the project, we want to put our attention on the future perspective of the project. As we know software development process begin with the interaction with the client. Clients mayn't have clear understanding of his need.

The improvement of software is not a rare scene for software industry. Such are of improvement in software is called the adaptive maintenance. In this type of maintenance we enhance our software.

In our project we used C++ language and its file-handling feature of C++. Our project doesn't also support multiple clients. But the scope of the project can be dramatically enhanced when we make this project online.

4.8 USER MANUAL

A user guide, a reference manual and often a tutorial are part of the documentation. These documentations are highly structured and can be written following precise guidelines.

User manual is basically provided for the user's convenience, as user doesn't know about the new system. So to operate the software user is given a training manual. The training manual describes simply the entire flow of the system, which helps the user to operate the system conveniently, and efficiency.

REFERENCES

1. Jamie Jaworsky 'Visual Basic 6.0' Techmedia.
2. Elmasri Navathe 'Fundamentals of database systems'.
3. Alam, M. (2011). Online Banking (1st ed.). New Delhi. LAP LAMBERT Academic Publishing, ISBN: 978-620-3-86302-4, DOI: 10.6084/m9.figshare.14612127.
4. Alam, M. (2012). Electronic Ticket Machine (1st ed.). New Delhi. LAP LAMBERT Academic Publishing, ISBN: 978-620-3-86332-1, DOI: 10.6084/m9.figshare.14661354.
5. Alam M, Khan AH, Khan IR (2016). Swarm intelligence in MANETS: a survey. Int J Emerg Res Manag Technol 5(5):141–150.
6. Alam, M. (2013). Just Shop-Shopping (1st ed.). New Delhi. LAP LAMBERT Academic Publishing, ISBN: 978-620-3-58124-9, DOI: 10.6084/m9.figshare.14431382.
7. Alam, M. (2013). Core ePortal (1st ed.). New Delhi. Glasstree Bookstore, ISBN: 978-1-6671-9827-9, DOI: 10.20850/9781667198279.
8. Alam, M., & Khan, M. (2013). E-Cops (1st ed.). New Delhi. LAP LAMBERT Academic Publishing, ISBN: 978-620-3-86368-0, DOI: 10.6084/m9.figshare.14662479.
9. E. Balaguruswamy, Programming in Turbo C++, TATA Mc Graw Hill.
10. Sumita Arora, Computer Science with C++, Text Book for Class XII.
11. Sumita Arora, Computer Science with C++, Text Book for Class XI.
12. Robert Lafore, Galgotia.
13. Alam, Mehtab. (2014). Steganography. DOI: 10.6084/m9.figshare.14662680.
14. Alam, M. (2016). Applicability of Swarm Intelligence in Mobile Ad Hoc Network (1st ed.). New Delhi. LAP LAMBERT Academic Publishing, ISBN: 978-620-3-57426-5, DOI: 10.6084/m9.figshare.14313548.
15. Kranacher, M. 2011. 2011 payroll tax holiday: Time to 'roll out the barrel'? The CPA Journal (January).
16. www.google.com
17. www.wikipedia.com
18. www.github.com

PART OF THE SOURCE CODE

```
*****
//      PROJECT "PAYROLL MANGEMENT SYSTEM"
//*****

//*****
//      INCLUDED HEADER FILES
//*****

#include<iostream.h>
#include<stream.h>
#include<proces.h>
#include<string.h>
#include<stdlib.h>
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<dos.h>
#include<stdlib.h>

//*****
// THIS CLASS CONTAINS ALL THE DRAWING FUNCTIONS
//*****

class LINES
{
    public :
        Void LINE_HOR(int, int, int, char) ;
        Void LINE_VER(int, int, int, char) ;
        Void BOX(int,int,int,int,char) ;
        void CLEARUP(void) ;
        Void CLEARDOWN(void) ;
};

//*****
// THIS CLASS CONTROLS ALL THE FUNCTIONS IN THE MENU
//*****
```

```

class MENU
{
    public :
        void MAIN_MENU(void) ;

    private :
        void EDIT_MENU(void) ;
        void INTRODUCTION(void) ;
};

//*****
// THIS CLASS CONTROL ALL THE FUNCTIONS RELATED TO EMPLOYEE
//*****

class EMPLOYEE
{
    public :
        void NEW_EMPLOYEE(void) ;
        void MODIFICATION(void) ;
        void DELETION(void) ;
        void DISPLAY(void) ;
        void LIST(void) ;
        void SALARY_SLIP(void) ;

    private :
        void ADD_RECORD(int, char[], char[], char[], int, int, int, char[], char, char,
char, float, float) ;
        void MODIFY_RECORD(int, char [], char [], char [], char [], char, char, char,
float, float) ;
        void DELETE_RECORD(int) ;
        int LASTCODE(void) ;
        int CODEFOUND(int) ;
        int RECORDNO(int) ;
        int FOUND_CODE(int) ;
        void DISPLAY_RECORD(int) ;
        int VALID_DATE(int, int, int) ;

        int code, dd, mm, yy ;
        char name[26], address[31], phone[10], desig[16] ;
        char grade, house, convense ;
        float loan, basic ;
};

//*****
// THIS FUNCTION CONTROL ALL THE FUNCTIONS IN THE MAIN MENU
//*****

void MENU::MAIN_MENU(void)
{
    char ch ;
    LINES L ;

```

```

L.CLEARUP() ;
while (1)
{
    clrscr();
    L.BOX(28,7,51,9,218) ;
    L.BOX(10,5,71,21,218) ;
    L.BOX(11,6,70,20,219) ;
    gotoxy(31,8);
    cout <<"WINTECH COMPUTERS" ;
    gotoxy(30,11) ;
    cout <<"1: NEW EMPLOYEE" ;
    gotoxy(30,12) ;
    cout <<"2: DISPLAY EMPLOYEE" ;
    gotoxy(30,13) ;
    cout <<"3: LIST OF EMPLOYEES" ;
    gotoxy(30,14) ;
    cout <<"4: SALARY SLIP" ;
    gotoxy(30,15) ;
    cout <<"5: EDIT" ;
    gotoxy(30,16) ;
    cout <<"0: QUIT" ;
    gotoxy(30,18) ;
    cout <<"ENTER YOUR CHOICE : " ;
    ch = getch() ;
    if (ch == 27 || ch == '0')
        break ;
    else
    if (ch == '1')
    {
        EMPLOYEE E ;
        E.NEW_EMPLOYEE() ;
    }
    else
    if (ch == '2')
    {
        EMPLOYEE E ;
        E.DISPLAY() ;
    }
    else
    if (ch == '3')
    {
        EMPLOYEE E ;
        E.LIST() ;
    }
    else
    if (ch == '4')
    {
        EMPLOYEE E ;
        E.SALARY_SLIP() ;
    }
    else

```

```

        if (ch == '5')
            EDIT_MENU() ;
    }
    L.CLEARUP() ;
}

//*****
// THIS FUNCTION CONTROL ALL THE FUNCTIONS IN THE EDIT MENU
//*****

void MENU::EDIT_MENU(void)
{
    char ch ;
    LINES L ;
    L.CLEARDOWN() ;
    while (1)
    {
        clrscr() ;
        L.BOX(28,8,49,10,218) ;
        L.BOX(10,5,71,21,218) ;
        L.BOX(11,6,70,20,219) ;
        gotoxy(31,9) ;
        cout <<"E D I T M E N U" ;
        gotoxy(30,13) ;
        cout <<"1: DELETE RECORD" ;
        gotoxy(30,14) ;
        cout <<"2: MODIFY RECORD" ;
        gotoxy(30,15) ;
        cout <<"0: EXIT" ;
        gotoxy(30,17) ;
        cout <<"ENTER YOUR CHOICE : " ;
        ch = getch() ;
        if (ch == 27 || ch == '0')
            break ;
        else
            if (ch == '1')
            {
                EMPLOYEE E ;
                E.DELETION() ;
            }
            else
                if (ch == '2')
                {
                    EMPLOYEE E ;
                    E.MODIFICATION() ;
                }
            }
        L.CLEARDOWN() ;
    }
}

```

```

/*****
// THIS FUNCTION DRAWS THE HORIZONTA L LINE
*****/

```

```

void LINES :: LINE_HOR(int column1, int column2, int row, char c)
{
    for ( column1; column1<=column2; column1++ )
    {
        gotoxy(column1,row) ;
        cout <<c ;
    }
}

```

```

/*****
// THIS FUNCTION DRAWS THE VERTICAL LINE
*****/

```

```

void LINES :: LINE_VER(int row1, int row2, int column, char c)
{
    for ( row1; row1<=row2; row1++ )
    {
        gotoxy(column,row1) ;
        cout <<c ;
    }
}

```

```

/*****
// THIS FUNCTION DRAWS THE BOX
*****/

```

```

void LINES :: BOX(int column1, int row1, int column2, int row2, char c)
{
    char ch=218 ;
    char c1, c2, c3, c4 ;
    char l1=196, l2=179 ;
    if (c == ch)
    {
        c1=218 ;
        c2=191 ;
        c3=192 ;
        c4=217 ;
        l1 = 196 ;
        l2 = 179 ;
    }
    else
    {

```

```

        c1=c ;
        c2=c ;
        c3=c ;
        c4=c ;
        l1 = c ;
        l2 = c ;
    }
    gotoxy(column1,row1) ;
    cout <<c1 ;
    gotoxy(column2,row1) ;
    cout <<c2 ;
    gotoxy(column1,row2) ;
    cout <<c3 ;
    gotoxy(column2,row2) ;
    cout <<c4 ;
    column1++ ;
    column2-- ;
    LINE_HOR(column1,column2,row1,l1) ;
    LINE_HOR(column1,column2,row2,l1) ;
    column1-- ;
    column2++ ;
    row1++ ;
    row2-- ;
    LINE_VER(row1,row2,column1,l2) ;
    LINE_VER(row1,row2,column2,l2) ;
}

//*****
// THIS FUNCTION CLEAR THE SCREEN LINE BY LINE UPWARD
//*****

void LINES :: CLEARUP(void)
{
    for (int i=25; i>=1; i--)
    {
        delay(20) ;
        gotoxy(1,i) ; clrhol() ;
    }
}

//*****
// THIS FUNCTION CLEAR THE SCREEN LINE BY LINE DOWNWORD
//*****

void LINES :: CLEARDOWN(void)
{
    for (int i=1; i<=25; i++)
    {
        delay(20) ;
        gotoxy(1,i) ; clrhol() ;
    }
}

```

```

    }
}

```

```

//*****
// THIS FUNCTION ADDS THE GIVEN DATA IN THE EMPLOYEE'S FILE
//*****

```

```

void EMPLOYEE :: ADD_RECORD(int ecode, char ename[26], char eaddress[31], char ephone[10], int
d, int m, int y, char edesig[16], char egrade, char ehous, char econv, float eloan, float ebasic)
{
    fstream file ;
    file.open("EMPLOYEE.DAT", ios::app) ;
    code = ecode ;
    strcpy(name,ename) ;
    strcpy(address,eaddress) ;
    strcpy(phone,ephone) ;
    dd = d ;
    mm = m ;
    yy = y ;
    strcpy(desig,edesig) ;
    grade = egrade ;
    house = ehous ;
    convense = econv ;
    loan = eloan ;
    basic = ebasic ;
    file.write((char *) this, sizeof(EMPLOYEE)) ;
    file.close() ;
}

```

```

//*****
// THIS FUNCTION MODIFY THE GIVEN DATA IN THE
// EMPLOYEE'S FILE
//*****

```

```

void EMPLOYEE :: MODIFY_RECORD(int ecode, char ename[26], char eaddress[31], char ephone[10],
char edesig[16], char egrade, char ehous, char econv, float eloan, float ebasic)
{
    int recno ;
    recno = RECORDNO(ecode) ;
    fstream file ;
    file.open("EMPLOYEE.DAT", ios::out | ios::ate) ;
    strcpy(name,ename) ;
    strcpy(address,eaddress) ;
    strcpy(phone,ephone) ;
    strcpy(desig,edesig) ;
    grade = egrade ;
    house = ehous ;
    convense = econv ;
    loan = eloan ;
}

```



```

        basic = ebasic ;
        int location ;
        location = (recno-1) * sizeof(EMPLOYEE) ;
        file.seekp(location) ;
        file.write((char *) this, sizeof(EMPLOYEE)) ;
        file.close() ;
    }

```

```

/*****
// THIS FUNCTION DELETE THE RECORD IN THE EMPLOYEE FILE
// FOR THE GIVEN EMPLOYEE CODE
*****/

```

```

void EMPLOYEE :: DELETE_RECORD(int ecode)
{
    fstream file ;
    file.open("EMPLOYEE.DAT", ios::in) ;
    fstream temp ;
    temp.open("temp.dat", ios::out) ;
    file.seekg(0,ios::beg) ;
    while (!file.eof())
    {
        file.read((char *) this, sizeof(EMPLOYEE)) ;
        if (file.eof())
            break ;
        if (code != ecode)
            temp.write((char *) this, sizeof(EMPLOYEE)) ;
    }
    file.close() ;
    temp.close() ;
    file.open("EMPLOYEE.DAT", ios::out) ;
    temp.open("temp.dat", ios::in) ;
    temp.seekg(0,ios::beg) ;
    while (!temp.eof())
    {
        temp.read((char *) this, sizeof(EMPLOYEE)) ;
        if ( temp.eof() )
            break ;
        file.write((char *) this, sizeof(EMPLOYEE)) ;
    }
    file.close() ;
}

```

```

        temp.close() ;
    }

```

```

//*****
// THIS FUNCTION RETURNS THE LAST EMPLOYEE'S CODE
//*****

```

```

int EMPLOYEE :: LASTCODE(void)
{
    fstream file ;
    file.open("EMPLOYEE.DAT", ios::in) ;
    file.seekg(0,ios::beg) ;
    int count=0 ;
    while (file.read((char *) this, sizeof(EMPLOYEE)))
        count = code ;
    file.close() ;
    return count ;
}

```

```

//*****
// THIS FUNCTION RETURNS 0 IF THE GIVEN CODE NOT FOUND
//*****

```

```

int EMPLOYEE :: FOUND_CODE(int ecode)
{
    fstream file ;
    file.open("EMPLOYEE.DAT", ios::in) ;
    file.seekg(0,ios::beg) ;
    int found=0 ;
    while (file.read((char *) this, sizeof(EMPLOYEE)))
    {
        if (code == ecode)
        {
            found = 1 ;
            break ;
        }
    }
    file.close() ;
    return found ;
}

```

```
}
```

```
/**
 * *****
 * // THIS FUNCTION RETURNS RECORD NO. OF THE GIVEN CODE
 * *****
 */
```

```
int EMPLOYEE :: RECORDNO(int ecode)
{
    fstream file ;
    file.open("EMPLOYEE.DAT", ios::in) ;
    file.seekg(0,ios::beg) ;
    int recno=0 ;
    while (file.read((char *) this, sizeof(EMPLOYEE)))
    {
        recno++ ;
        if (code == ecode)
            break ;
    }
    file.close() ;
    return recno ;
}
```


More
Books!



Buy your books fast and straightforward online - at one of world's fastest growing online book stores! Environmentally sound due to Print-on-Demand technologies.

Buy your books online at

www.morebooks.shop

Kaufen Sie Ihre Bücher schnell und unkompliziert online – auf einer der am schnellsten wachsenden Buchhandelsplattformen weltweit! Dank Print-On-Demand umwelt- und ressourcenschonend produziert.

Bücher schneller online kaufen

www.morebooks.shop

KS OmniScriptum Publishing
Brivibas gatve 197
LV-1039 Riga, Latvia
Telefax: +371 686 204 55

info@omniscryptum.com
www.omniscryptum.com

OMNIScriptum



