

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 General introduction**

Employee Management System Project is developed Using Visual Basic 6.0 and can get the documentation file (employee payroll and leave management) System must provide built-in feature of security and integrity of data without safe guard against unauthorized development access, fraud, even embellzzlement, fire and natural disasters, a system could be so well vulnerable as to threaten the survival of the organization .Marked Categories : employee, full project report on employee information management system, employee management system project documentation , employee management. To do an educate job on security a system analyst must analyze the risk exposure and cost and specify measures such as password and encryption to provide protection. In addition backup copy of the software and recovery restart procedure must be available when needed. a disaster/recovery that has management support should also be prepared. Then no matter what the disaster, the firm can recover.

The trend of automating payroll and workforce management processes began during the 1970s when due to limited technology and Mainframe computer , companies were still relying on manual entry to conduct employee evaluation and to digitize reporting.

This system gave users the possibility to combine corporate data in real time, and regulate processes from a single mainframe environment. Many of today's popular HR systems still offer considerable ERP and payroll functionality.

## **1.2 Objectives:**

To show, add, edit and delete the data or information of the employee.

1. To make the organisation's employee information secure.
2. To make it easier and convenient than paper work.
3. To control the office management system in systematic way.

The objective of employee management system is to allow the administrator of any organization to edit and find out the details of a employee. It will also facilitate keeping all the records of employees. So all the information about a employee will be available in a few seconds.

Employee management system is intended to help any organization to maintain and manage its employee's personal data.

## **1.3 METHODOLOGY TO BE FOLLOWED:**

The scope of the project includes the following :

1. Any organization can use this system as it is not client centric.
2. All information updating can be done by this.
3. Deliver Electronic Workplace
4. Application Support & Maintenance after deployment to production
5. It is reusable in other management systems.
6. This system is user-friendly.
7. Fast data entry.

## **1.4 EXPECTED OUTCOMES:**

- 1.This is used to create a record of the employee.
- 2.It stores the salary of the employee and deductions are considered as per the holidays taken by the employee.
- 3.It is used to display all the records of the employee.

### 1.5 HARDWARE REQUIREMENTS:

🚦 Processor: Intel core i3/i4/i5/i6/i7

- Speed:3.20 GHz to 3.60 GHz

🚦 RAM:512MB RAM or more.

🚦 ROM:2 GB DDR3

### SOFTWARE REQUIREMENTS:

- Dev C/C++.
- Turbo C/C++.

Windows XP/Windows 7/Windows 8/Windows 10

## **CHAPTER 2**

### **DATA STRUCTURES**

#### **2.1 FILES**

In order to provide different details for each employee based on their employee number and account number there are 5 cases with different particulars.

Under the existing employee record, the administrator has to first register the employee name which will be unique for each employee. Then he will have to enter employee number then employee salary, then employee salary deductions and employee holidays given. After submitting these data, the administrator will be given option if he wants to delete any record or update or search and he can even display it.

- Doesn't require any extra hardware device
- Very fast and accurate
- No need of extra manual effort
- Proposed system provides domain login facility so no need to remember password and login id.
- Can store large amount of data

A well-conceived and implemented automated employee record system can reduce the costs of handling the paperwork associated with record keeping. When information from a employee record is requested, it is usually needed promptly. A administrator making decisions about a new employee needs the previous organisations records immediately to assign the employee to the appropriate programs and services. A well-designed employee record system allows for

timely retrieval of needed information in some important situations like conference, projects etc.

Data quality is basic to a well-designed employee record system. Having clearly defined data elements that are used consistently promotes data quality. Paper records have traditionally been considered accurate, although not necessarily complete or accessible. Maintaining data quality as information is shared, analyzed, and reported is a characteristic of a well-designed system. Accuracy is vital at every stage, from data collection, to entry, to maintenance in the system. Accuracy provides users with the confidence they will require to rely upon a employee record system.

An effective automated employee record system should provide the information required on request, easily, and without burdensome trial-and-error searching. From the first steps of designing the system, the queries that will follow are anticipated and accommodated. Similar to an office filing system, the adequacy of a employee record system is often judged by how much time and effort are required to find and retrieve information. Therefore, a key part of the design of a employee record system is its process for access, retrieval, and reporting.

## CHAPTER 3

### 3.1 Flow Chart Diagram

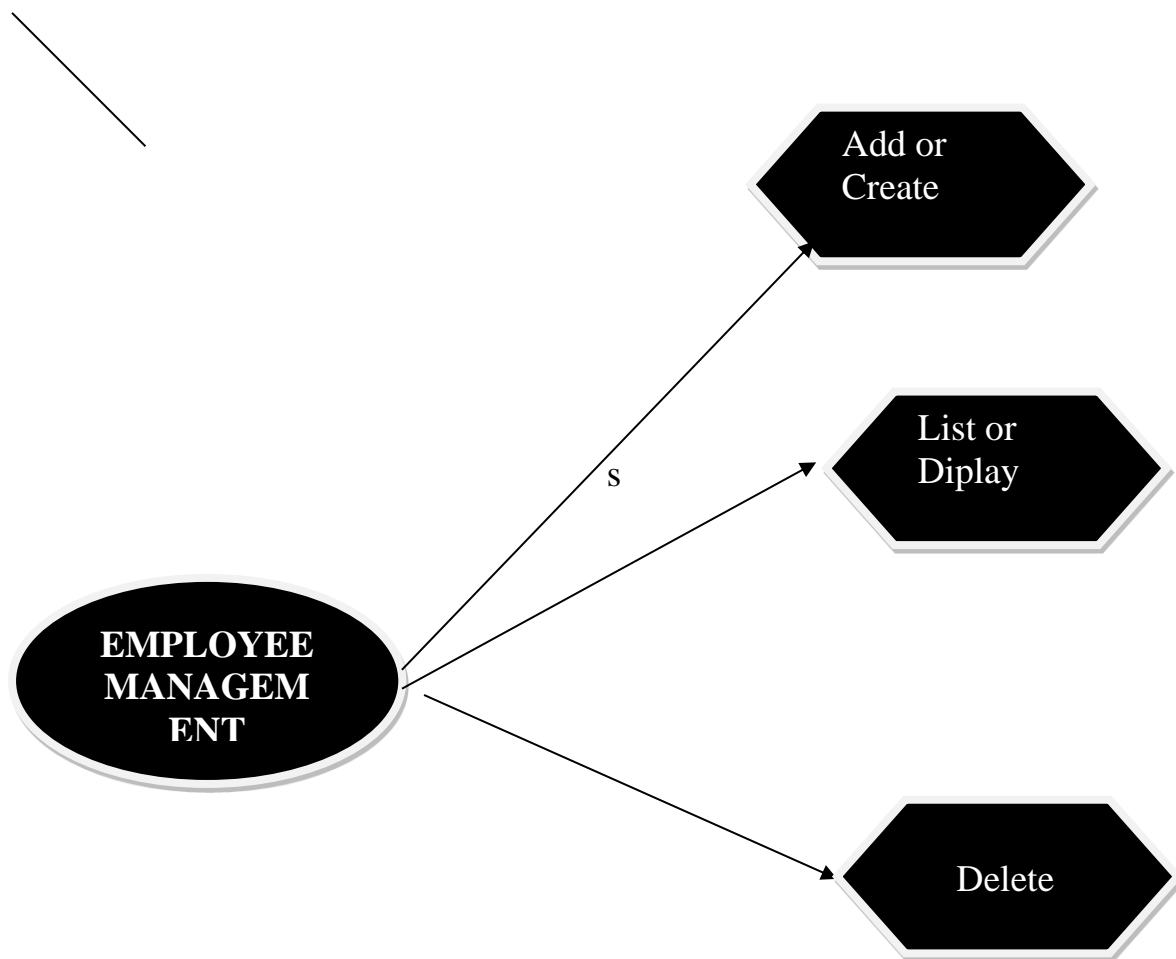


FIG 3.1

### 3.2Algorithm

#### **Pseudocode 1: to create record**

1. STEP 1:create a new record.
2. STEP 2:print the record number,employee name, employee number , employee salary, employee salary deductions, and number of holidays
3. STEP 3:employee will be added to the record

#### **Pseudocode 2: to display**

1. STEP 1:check if you don't want to add another record
2. STEP 2:if no then print employee account number , employee name , employee number , employee salary , employee deductions , holidays
3. STEP 4: record will be displayed

#### **Pseudocode 3:deletion of an existing record**

1. STEP 1:once you go for the option 3
2. STEP 2:displays all the existing records
3. STEP 3: enter employee account number to delete, record will be deleted

## CHAPTER 4

### IMPLEMENTATION:

```
/// List of library functions

#include <stdio.h> ///for input output functions like printf, scanf

#include <stdlib.h>

#include <string.h> ///string operations


void flush()

{

    int c;

    while ((c = getchar()) != '\n' && c != EOF)

        ;

}


/** Main function started */


int main()

{

    FILE *fp, *ft; /// file pointers

    char another, choice;


    /** structure that represent a employee */

    struct emp
```



```
{
    int empno;

    char name[40]; ///name of employee

    int ded;    /// employee deductions

    int hol;    /// hol of employee

    float bs;    /// basic salary of employee
};

struct emp e; /// structure variable creation

char empname[40]; /// string to store name of the employee

long int recsize; /// size of each record of employee

/** open the file in binary read and write mode
 * if the file EMP.DAT already exists then it open that file in read write mode
 * if the file doesn't exit it simply create a new copy
 */

fp = fopen("EMP.DAT", "rb+");
if (fp == NULL)
{
    fp = fopen("EMP.DAT", "wb+");
    if (fp == NULL)
    {
```

```
    printf("Connot open file");
    exit(1);
}
}

// sizeo of each record i.e. size of structure variable e
recsize = sizeof(e);

// infinite loop continues untile the break statement encounter
while (1)
{
    printf("1. Add Record\n");    /// option for add record
    printf("2. List Records\n");  /// option for showing existing record
    printf("3. Delete Records\n"); /// option for deleting record
    printf("4. Exit\n");          /// exit from the program
    printf("Your Choice: ");      /// enter the choice 1, 2, 3, 4
    fflush(stdin);                /// flush the input buffer
    scanf("\n%c", &choice);      /// get the input from keyboard
    switch (choice)
    {
        case '1':                /// if user press 1

            fseek(fp, 0, SEEK_END); /// search the file and move cursor to end of
the file

            /// here 0 indicates moving 0 distance from the end of the
file
```

```
another = 'y';

while (another == 'y')
{ /// if user want to add another record

    flush();

    printf("\nEnter Employee name: ");

    fgets(e.name, 40, stdin);

    printf("\nEnter Employee Number: ");

    scanf("%d", &e.empno);

    printf("\nEnter basic salary: ");

    scanf("%f", &e.bs);

    printf("\nEnter Deductions: ");

    scanf("%d", &e.ded);

    printf("\nEnter holidays: ");

    scanf("%d", &e.hol);


    fwrite(&e, reccount, 1, fp); /// write the record in the file


    printf("\nAdd another record(y/n) ");

    fflush(stdin);

    scanf("\n%c", &another);

}

break;

case '2':

    rewind(fp); ///this moves file cursor to start of the file
```

```
while (fread(&e, reccsize, 1, fp) == 1)

{ /// read the file and fetch the record one record per fetch


    printf("%s %d %.2f %d %d\n", e.name, e.empno, e.bs, e.ded, e.hol);
/// print the name, hol and basic salary

}

break;


case '3':

    another = 'y';

    while (another == 'y')

    {

        flush();

        printf("\nEnter name of employee to delete: ");

        fgets(empname, 40, stdin);

        ft = fopen("Temp.dat", "wb"); /// create a intermediate file for
temporary storage

        rewind(fp);                /// move record to starting of file

        while (fread(&e, reccsize, 1, fp) == 1)

        { /// read all records from file

            if (strcmp(e.name, empname) != 0)

            {

                /// if the entered record match

                fwrite(&e, reccsize, 1, ft); /// move all records except the one that
is to be deleted to temp file

            }

        }

    }

}
```

```
    }

    fclose(fp);

    fclose(ft);

    remove("EMP.DAT");          /// remove the orginal file

    rename("Temp.dat", "EMP.DAT"); /// rename the temp file to original
file name

    fp = fopen("EMP.DAT", "rb+");

    printf("Delete another record(y/n)");

    fflush(stdin);

    scanf("\n%c", &another);

}

break;

case '4':

    fclose(fp); /// close the file

    exit(0);  /// exit from the program

}

}
```

## CHAPTER 5:

### RESULTS:

It displays the options to choose

```
C:\TURBOC3\BIN>TC
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice: _
```

## EMPLOYEE MANAGEMENT SYSTEM

---

It store the details

```
C:\TURBOC3\BIN>TC
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice: 1

Enter Employee name: bhargavi

Enter Employee Number: 1

Enter basic salary: 10000

Enter Deductions: 3000

Enter holidays: 2

Add another record(y/n)
```

## EMPLOYEE MANAGEMENT SYSTEM

---

In this it asks for employee details

```
Your Choice: 1
Enter Employee name: bhargavi
Enter Employee Number: 1
Enter basic salary: 10000
Enter Deductions: 3000
Enter holidays: 2
Add another record(y/n) y
Enter Employee name: ramya
Enter Employee Number: 2
Enter basic salary: 12000
Enter Deductions: 4000
Enter holidays: 2
Add another record(y/n) n_
```



## EMPLOYEE MANAGEMENT SYSTEM

---

Again it asks for other options

```
Enter Employee Number: 2
Enter basic salary: 12000
Enter Deductions: 4000
Enter holidays: 2
Add another record(y/n) n
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice: 2
sana
  12 40000.00 4000 10
bhargavi
  1 10000.00 3000 2
ramya
  2 12000.00 4000 2
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice:
```

Here it displays the record details

```
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice: 3

Enter name of employee to delete: ramya
Delete another record(y/n)n
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice: 2
sana
12 40000.00 4000 10
1. Add Record
2. List Records
3. Delete Records
4. Exit
Your Choice:
```

## CHAPTER 6

### CONCLUSION

Simplicity is never simple. As we have seen in this project, the process of creating a user-friendly and straightforward platform that facilitates the administrator's job is one filled with complexity. From understanding user requirements to system design and finally system prototype and finalization, every step requires in-depth understanding and commitment towards achieving the objectives of the project.

The system prototype demonstrates easy navigation and data are stored in a systematic way. Hence to enable the smooth and efficient functioning of system and to update all the records of the new employees with the existing ones, Systematic management of data school employee management system is necessary in every organization.

### References:

#### Books

:Dr.PADMA REDDY- Data structures

:Dr.DENNIS RICHIE- C learning