**DIRE DAWA UNIVERSITY**

**INSTITUTE OF TECHNOLOGY**



**DEPARTMENT OF SOFTWARE ENGINEERING**

**COMPUTER PROGRAMMING PROJECT**

**EMPLOYEE MANAGEMENT SYSTEM**

**GROUP 4**

**GROUP MEMBERS ID**

1. Dano Dula 1302484
2. Mamush Fikadu 1303319
3. Abdurahman Sherefa 1301902
4. Abdulhafiz Mohammedhussen 1301893
5. Ebisa Shugi 1302577
6. Bayisa Bekele 1302227
7. Nebiyu Tilikalem 1303604

Submitted to: Ins. Seid M.

Submission Date: 06/09/2014

## Dire Dawa University Employee Management System

**Dire Dawa University** Employee Management System is a software built to handle the primary employee management of the company. It help the company keep track of all the employees and their records. It is used to manage the company using computerized system.

Aim of Employee’s Management System:

* Enter information of multiple employees
* Insert New Employee information
* Delete The Employee information.
* Search Employee record.

Data of the Employee’s:

* Name
* Employee ID
* Title
* Years of Experience
* Age

Approach:

1. For storing the data of the employee, create a user define datatype which will store the information regarding Employee. Below is the declaration of the data type:

struct employee {

string name;

long int code;

string designation;

int exp;

int age;

};

1. Entering Multiple Information: For Entering information of multiple employees, the idea is to use the array of the above struct datatype which will use to store the information regarding employee. First it asks the number of information of employees and it can take a maximum of 20 different data, then we can store information of every single employee one by one. For storing information at index ‘i’ the data is stored as:

struct employee emp[10];

emp[i].name = "Abebe"

emp[i].code = "12345"

emp[i].designation = "Salesman"

emp[i].exp = 5

emp[i].age = 10

1. Deleting in the record: Since we are using array to store the data, therefore to delete the data at any index shift all the data at that index by 1 and delete the last data of the array by decreasing the size of array by 1.
2. Searching in the record: For searching in the record based on any parameter, the idea is to traverse the data and if at any index the value parameters matches with the record stored, print all the information of that employee.

**BENEFITS OF THIS SYSTEM:-**

* **This system wii reduce the complexcity of employee management.**
* **It will reduce searching time.**
* **It can be easily handeled by the person who have elementary knowledge of computer because it provides an user friendly environment.**
* **It’s hardware and software configuration is not very costly that means**

## CONCLUSION

This project is developed using c ++ fully meets the objective of the system which it has been developed. This project is used for computerizing employee management work in Dire Dawa University. The software keeps record of employee’s name , age, years of experience, Title, too. The software is capable of easy storage of information related to employee through database.

#include <bits/stdc++.h>

#include <iostream>

#define max 20

using namespace std;

struct employee {

string name;

long int code;

string designation;

int exp;

int age;

};

int num;

void showMenu();

/\* Array of Employees to store the data in the form of the Structure of the Array \*/

employee emp[max];

// Function to Enter Multiple Informations

void build()

{

cout << "Enter The Number of Employees to be Hired"<<endl;

cout << "Maximum Entries should be "<< max << "\n";

cin >> num;

if (num > 20) {

cout << "Maximum number of Entries are 20\n";

num = 20;

}

cout << "Enter the following data:\n";

for (int i = 0; i < num; i++) {

cout << "Name ";

cin >> emp[i].name;

cout << "Employee ID ";

cin >> emp[i].code;

cout << "Title ";

cin >> emp[i].designation;

cout << "Years of Experience ";

cin >> emp[i].exp;

cout << "Age ";

cin >> emp[i].age;

}

showMenu();

}

// Function to insert data if single employee

void insert()

{

if (num < max) {

int i = num;

num++;

cout << "Enter the information of the Employee\n";

cout << "Name ";

cin >> emp[i].name;

cout << "Employee ID ";

cin >> emp[i].code;

cout << "Title ";

cin >> emp[i].designation;

cout << "Years of Experience ";

cin >> emp[i].exp;

cout << "Age ";

cin >> emp[i].age;

}

else {

cout << "Employee Table Full\n";

}

showMenu();

}

// Function to delete record at index i

void deleteIndex(int i)

{

for (int j = i; j < num - 1; j++) {

emp[j].name = emp[j + 1].name;

emp[j].code = emp[j + 1].code;

emp[j].designation = emp[j + 1].designation;

emp[j].exp = emp[j + 1].exp;

emp[j].age = emp[j + 1].age;

}

return;

}

// Function to delete record

void deleteRecord()

{

cout << "Enter the Employee ID to Delete Record"<<endl;

int code;

cin >> code;

for (int i = 0; i < num; i++) {

if (emp[i].code == code) {

deleteIndex(i);

num--;

cout<<"Entry is deleted"<<endl;

break;

}

}

showMenu();

}

void searchRecord()

{

cout << "Enter the Employee ID to Search Record"<<endl;

int code;

cin >> code;

for (int i = 0; i < num; i++) {

if (emp[i].code == code) {

cout << "Name " << emp[i].name << "\n";

cout << "Employee ID "<< emp[i].code << "\n";

cout << "Designation "<< emp[i].designation << "\n";

cout << "Experience "<< emp[i].exp << "\n";

cout << "Age "<< emp[i].age << "\n\n\n";

break;

}

}

showMenu();

}

// Function to show menu

void showMenu()

{

cout << "-------------"

<< "Dire Dawa University Employee Management System------------\n\n";

cout << "Available Options:\n\n";

cout << "Enter Information of Multiple Employees (1)\n";

cout << "Insert New Employee (2)\n";

cout << "Delete Employee Information (3)\n";

cout << "Search Employee (4)\n";

cout << "Exit (5)\n";

int option;

cin >> option;

if (option == 1) {

build();

}

else if (option == 2) {

insert();

}

else if (option == 3) {

deleteRecord();

}

else if (option == 4) {

searchRecord();

}

else if (option == 5) {

return;

}

else {

cout << "Expected Options are 1/2/3/4/5";

showMenu();

}

}

int main()

{

showMenu();

return 0;

}