Telephone Directory   
maintenance

System

Date:-28-08-2019.



SDM COLLEGE OF ENGINEERING AND TECHNOLOGY , DHAVALAGIRI, DHARWAD – 580002

**Cse Department**

**Batch :-A2**

# Team Member details

|  |  |  |
| --- | --- | --- |
| Name | USN | Roll No |
| Shetty Ganeshprasad | 2SD17CS088 | 134 |
| Samarth Gaonkar | 2SD17CS080 | 131 |
| Nikhil Verneker | 2SD17CS053 | 120 |
| Niranjan kumar | 2SD17CS054 | 121 |

# TELEPHONE Directory maintenance system

|  |  |  |
| --- | --- | --- |
| Sr.no | TOPIC | Page no. |
| 1.0 | Project overview. | 3 |
| 1.1 | List of files supplied. | 3 |
| 1.2 | Different Modules used. | 3 |
| 2.0 | Login Screen | 4 |
| 2.1 | Main Menu | 4 |
| 3.0 | Department Maintenance Menu | 5 |
| 3.1 | Adding Department | 5 |
| 3.2 | Exceptions handled. (Avoiding Duplication in Department) | 5 |
| 3.3 | Department details display | 6 |
| 4.0 | Employee Maintenance Menu. | 6 |
| 4.1 | Adding Employees Details. | 7 |
| 4.1.1 | Exception handled while entering employee data | 7 |
| 4.2 | Printing Employees Details. | 9 |
| 4.3 | Deleting Employee from file | 10 |
| 5.0 | Telephone Directory Menu | 10 |
| 5.1 | Add Telephone number. | 11 |
| 5.2 | Telephone Enquiry Menu. | 12 |
| 5.2.1 | Telephone Enquiry Based on Name. | 12 |
| 5.2.2 | 2Telephone Enquiry Based on Telephone Number | 12 |
| 6 | Source Code of full project. | 13-30 |

## 1.0 Project overview.

|  |  |
| --- | --- |
|  | This Project create maintain and store details of Employees their department and there company assigned tele phone number and print its details on demand. |
|  |  |

# 1.1 list of files supplied

|  |  |  |
| --- | --- | --- |
| S.no | File Name | Description |
| 1 | FileFunctions.c | Source file which contains all the functions required to handle records in file. Functions also have documentation in function header comment blocks. |
| 2 | FileFunctions.h | Header file which contains the declaration of all the functions and constants in FileFunctions.c. Include this in any new source file that you write to use the functions. |
| 3 | DeptMaint.c | File containing code to add and view departments in the department file |
| 4 | TelDir.c | The main file of telephone directory system. Contains code for the main menu and department menu. |
| 5 | dept.txt | File which your program uses to store information about the departments. |
| 6 | emp.txt | File which your program uses to store information about employees. |

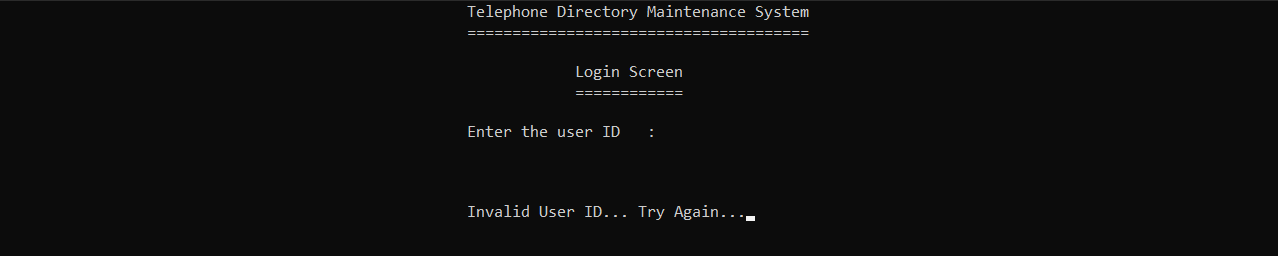
## 1.2 Different Modules used.

|  |  |
| --- | --- |
|  | This project is created using modular approach as demanded in Specs |

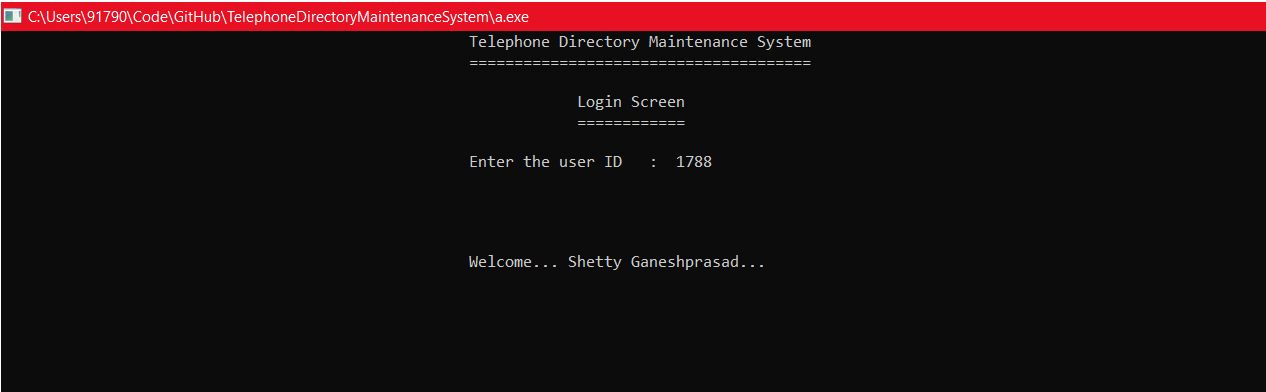
* FileFunctions.c (Main running module which will call other module based on user input).
* EmpMaint.c ( For employee maintenance. This module is called by “FileFunctions.c” module).
* DeptMaint.c (For Dept maint. This module is called by “FileFunctions.c” module ).
* TelDir.c (For maintaining Telephone directory called by “FileFunction.c” module).
* FileFunctions.h (For storing all different functions used.This header file will be imported.

## 2.0 Login Screen

|  |  |
| --- | --- |
|  | **Currently there are 4 logins of our teammate any login other than this is invalid login and no access to data.** |



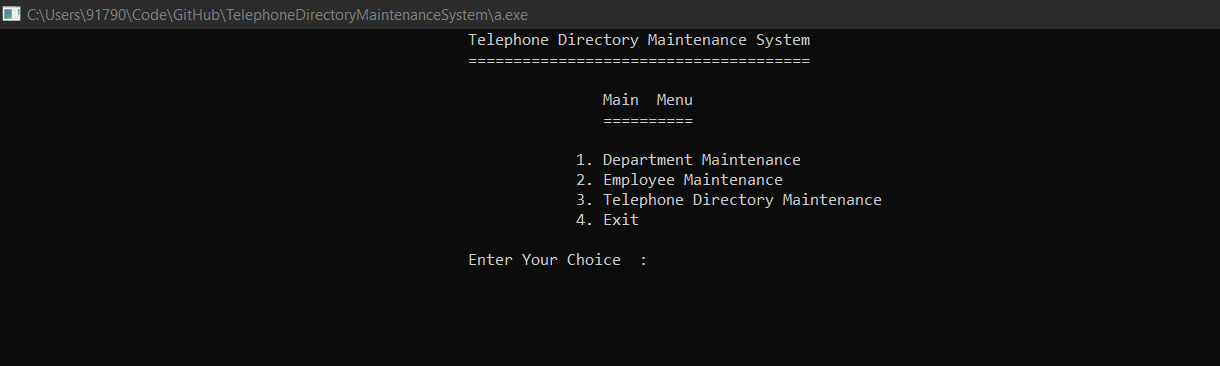
With correct Login no Welcome msg is shown for 3 seconds



After this screen programme redirect it to main menu.

## 2.1 Main Menu

|  |  |  |
| --- | --- | --- |
|  | **U.I of main menu** |  |



Simple Switch based menu is generated and programme wait for the user input.

Based on user input different programme is executed.

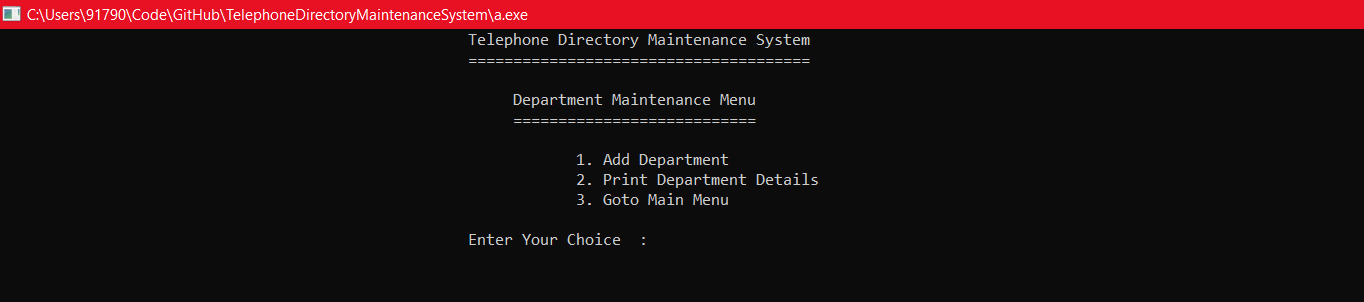
For example :- for I/P “1” “DeptMaint.exe” is executed.

For I/P “2” “EmpMaint.exe” is executed.

For I/P “3” “EmpMaint.exe” is executed. similarly I/P “4” for terminating prog

## 3.0 Department Maintenance Menu

|  |  |
| --- | --- |
|  | **U.I of main menu** |



Simple Switch based menu is generated and programme wait for the user input.

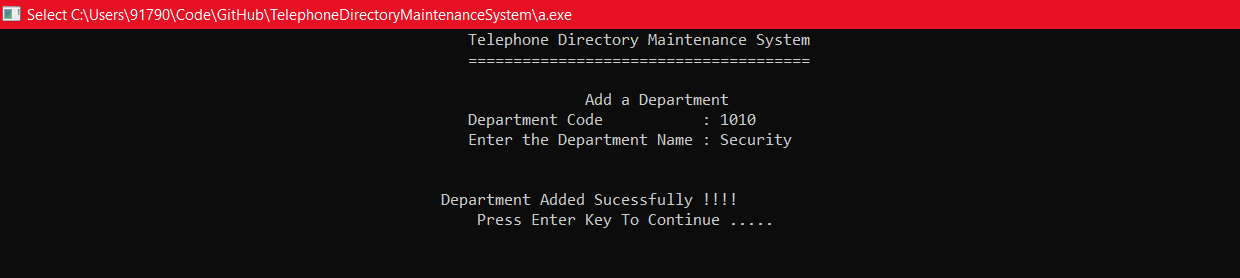
For example :- for I/P “1” Add new department (1.1)

For I/P “2” Printing Dept Details (1.2)

For I/P “3” Closing “DeptMaint.exe” and returning to main menu (0.)

## 3.1 Adding Department

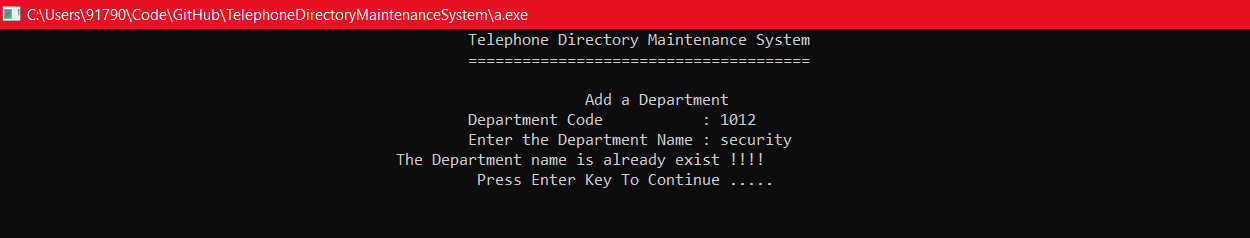
|  |  |
| --- | --- |
|  | **Screen when Dept is added successfully** |



Programme assign Department Code automatically and ask users input for Dept Name.

## 3.2 Exceptions handled. (Avoiding Duplication in Department)

|  |  |
| --- | --- |
|  | **Screen when Dept Name entered is already exixting** |

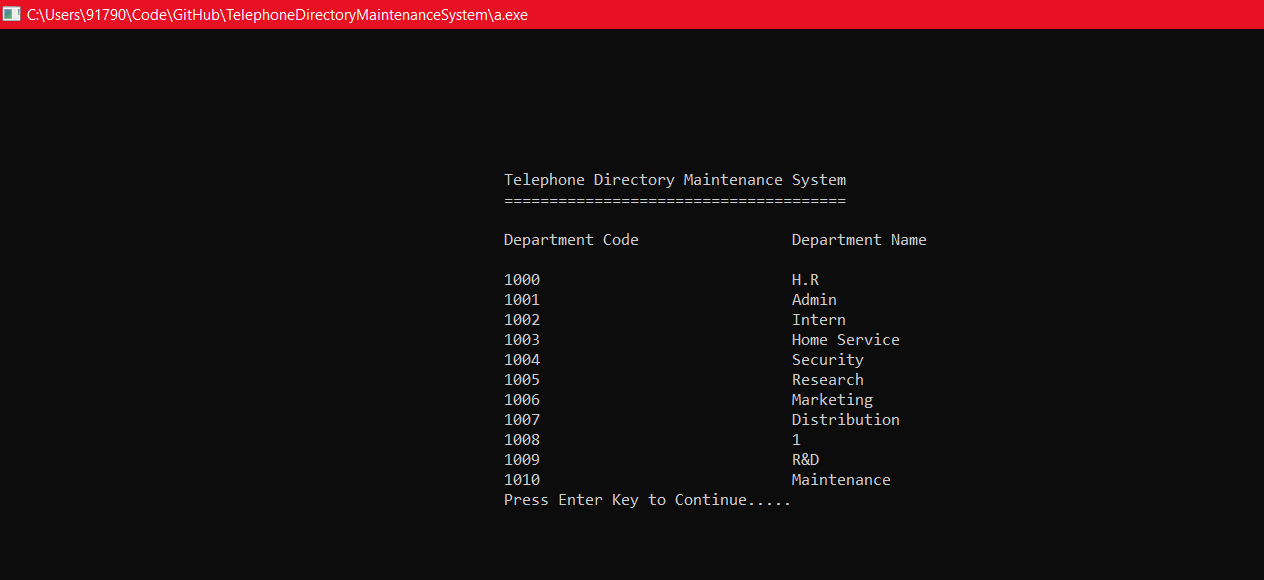


The code responsible for detecting Duplication is case Insensitive

(i.e:- Department named “CSE” and “cse” will be treated as same departments.

## 3.3 Department details display

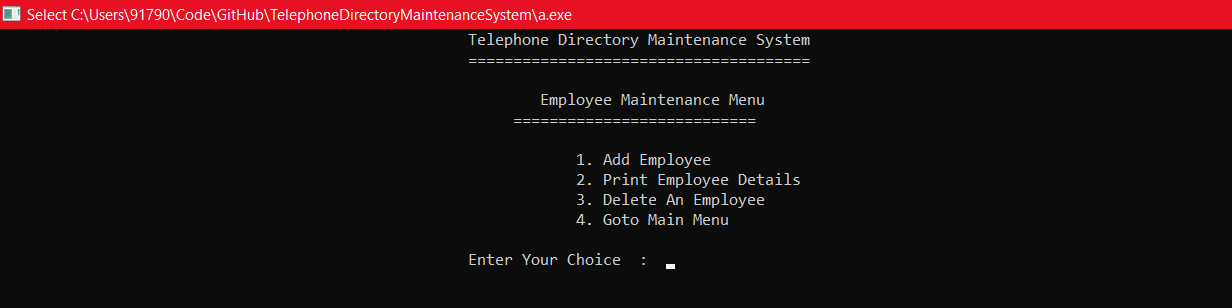
|  |  |
| --- | --- |
|  | **Screen when Dept details is displayed** |



Displays all the department’s Code and Department Nam

## 4.0 Employee Maintenance Menu.

|  |  |
| --- | --- |
|  | **Screen for displaying menu for employee maintenance.** |



Simple Switch based menu is generated and programme wait for the user input.

Based on user input different programme is executed.

For example :- for I/P “1” Add Employee.

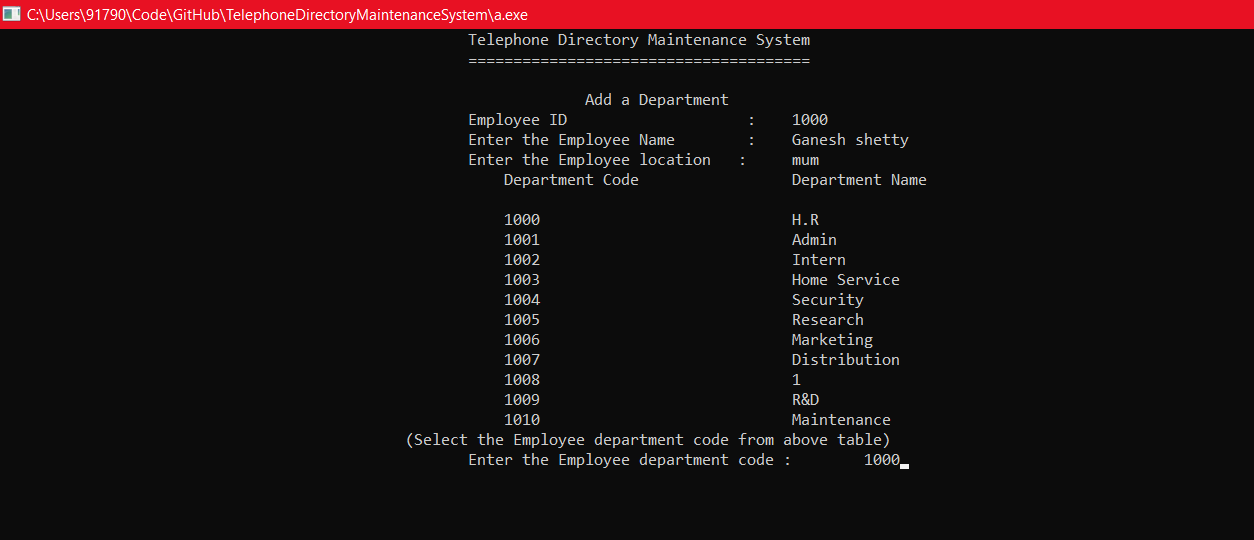
For I/P “2” printing Employee.

For I/P “3” Deleting Employee.

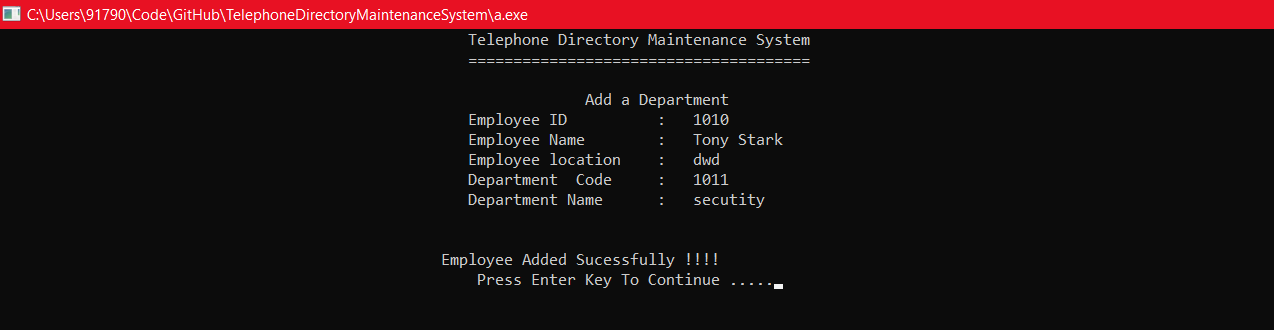
For I/P “4” for go to main menu.

## 4.1 Adding Employees Details.

|  |  |
| --- | --- |
|  | **Screen while Adding employee details** |

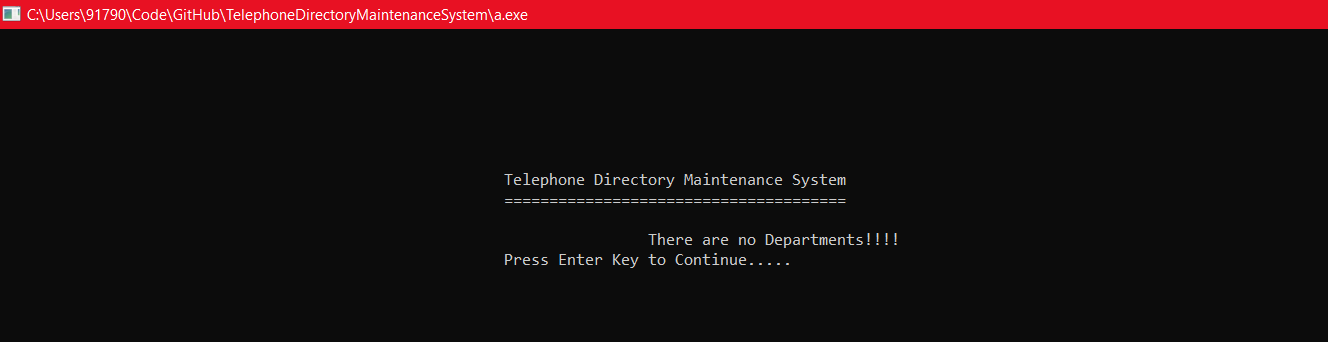


Once department No is given as input the Employee will be added to that department



Employee id is automatically assigned to employee. Name, location,and dept code is asked as input from user.

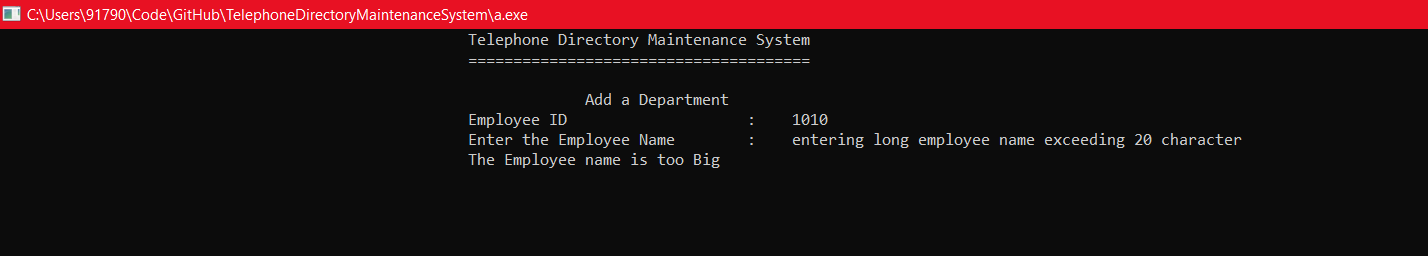
## 4.1.1Exceptions handled !! .(Entering Employee Without having Deptartment)



If the user wants to add new employees without creating a single department thwn this error message is shown stating “**There are no Department**”

## 4.1.2 Exceptions handled !! .(max character constraint in Name)

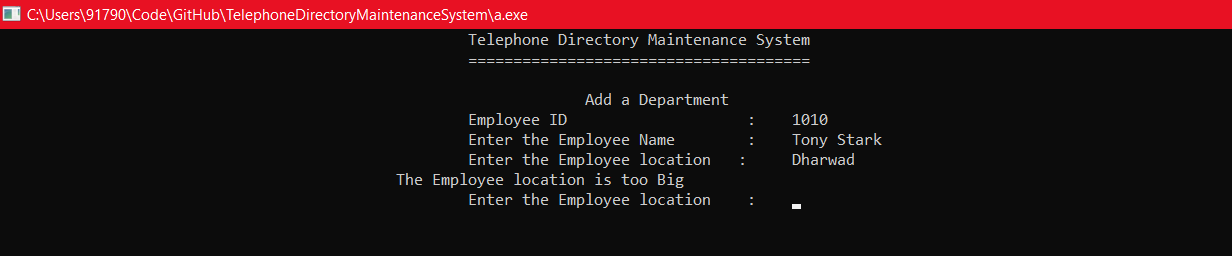
|  |  |
| --- | --- |
|  | **Exception for exceeding max limit size of details** |



Employee Name can be of max 25 character. Any entry more than that limit will not be accepted and “The Employee name is too big” error message is shown on screen.

## 4.1.3 Exceptions handled !! .(max character constraint in Address.)

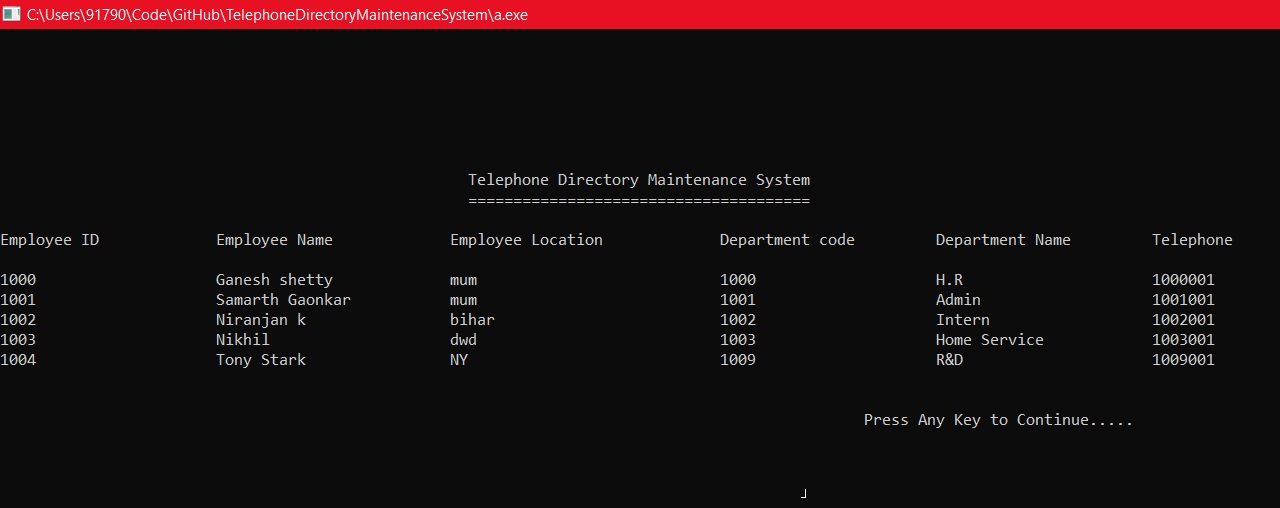
|  |  |
| --- | --- |
|  | **Exception for exceeding max limit size of details** |



Employee Address can be of max 5 character. Any entry more than that limit will not be accepted and “The Employee Location is too big” error message is shown on screen.

## 4.2 Printing Employees Details.

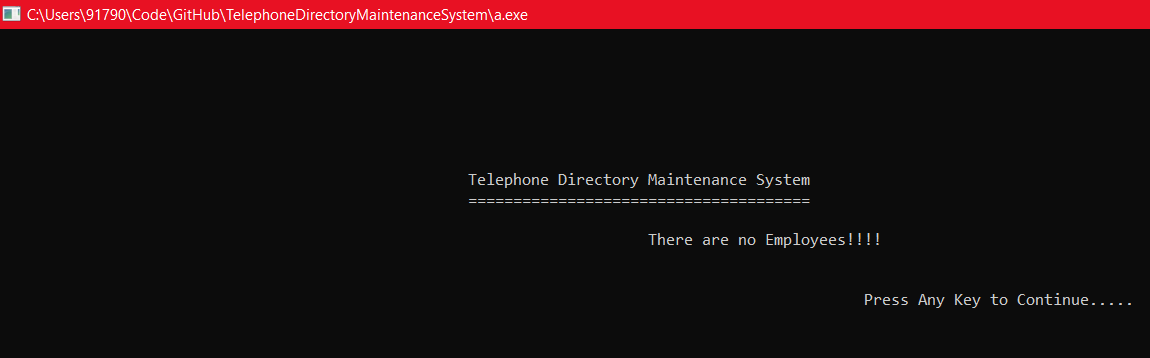
|  |  |  |
| --- | --- | --- |
|  |  | **Screen while Printing Employee details** |



All Employee Details are printed with their Name,Location,Dept code, Dept Name,Telephone.

## 4.2.1. Exception Handled !! (Printing Employees Details When no employee Entered)

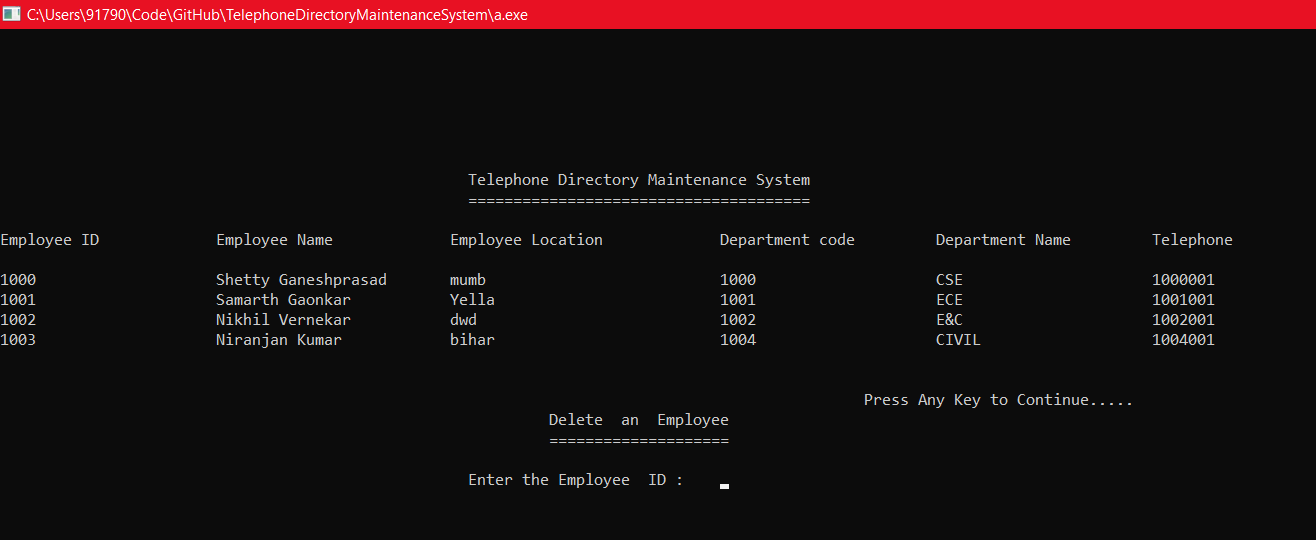
|  |  |
| --- | --- |
|  | **Error messages** |



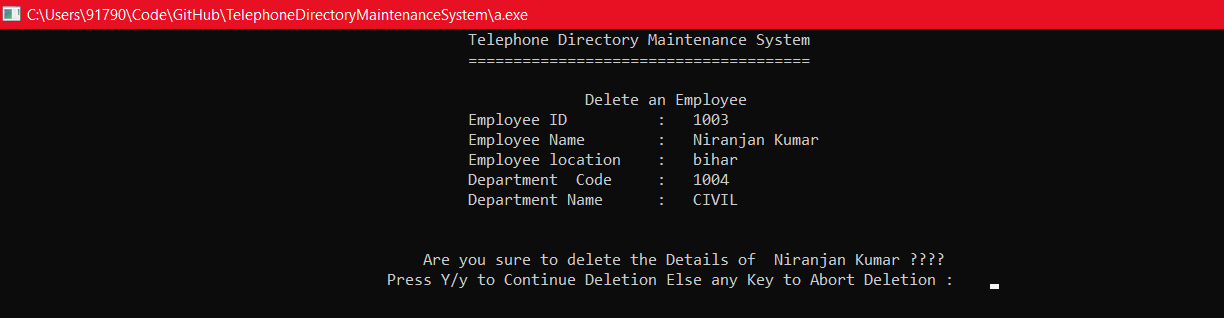
When no employee is added still userwant to print it.

## 4.3 Deleting Employee from file

|  |  |
| --- | --- |
|  | **Deleting screen for employe** |



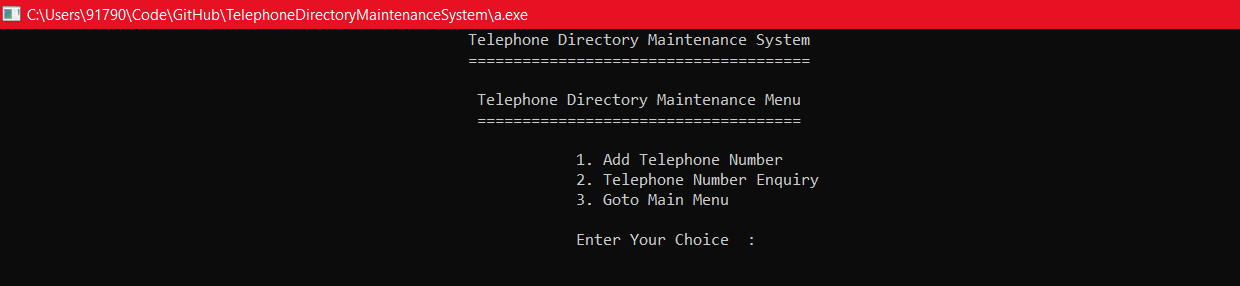
After selecting Employee id number



After Selecting “Y” the employee gets deleted.

## 5.0 Telephone Directory Menu

|  |  |
| --- | --- |
|  | **UI of telephone directory** |



Simple Switch based menu is generated and programme wait for the user input.

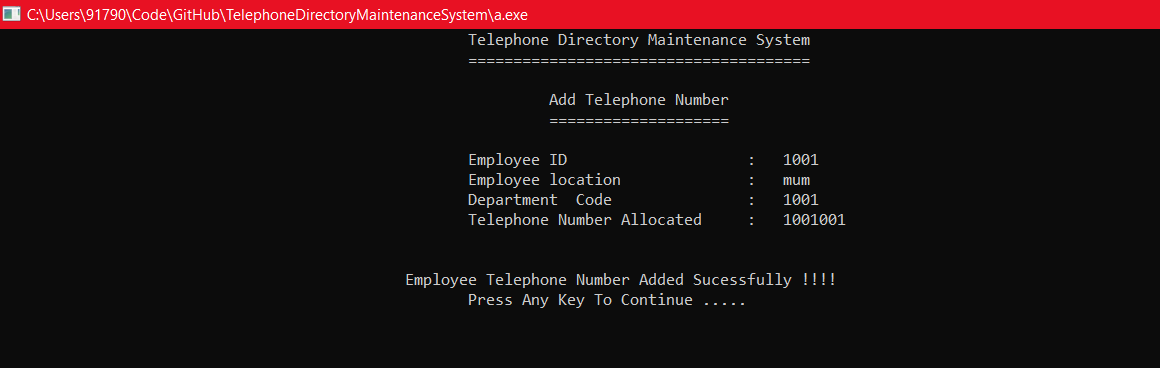
For example :- for I/P “1” Add new Telephone number for employee

For I/P “2” Enquiry for telephone Details ()

For I/P “3” Closing “EmpMaint.exe” and returning to main menu (0)

## 5.1 Add Telephone number.

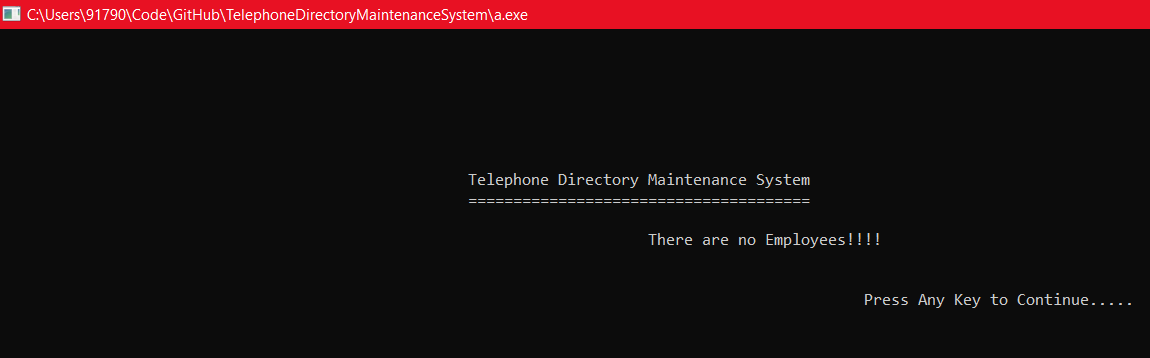
|  |  |
| --- | --- |
|  | **UI of telephone directory** |



Programme will wait for user input of Employee id and Assign the new telephone number for that employee. With some basic details of that employee.

## 5.1.1 Exceptions handled !! .(If Employee id is not valid .)

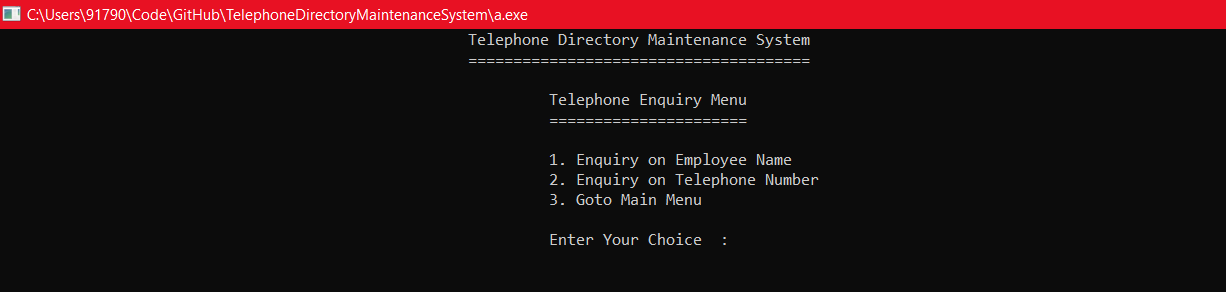
|  |  |
| --- | --- |
|  | **Exception for wrong Employee id** |



If the employee id entered is not valid this error screen will be shown

## 5.2 Telephone Enquiry Menu.

|  |  |
| --- | --- |
|  | **UI of Telephoe Enquiry menu** |

 Simple Switch based menu is generated and programme wait for the user input.

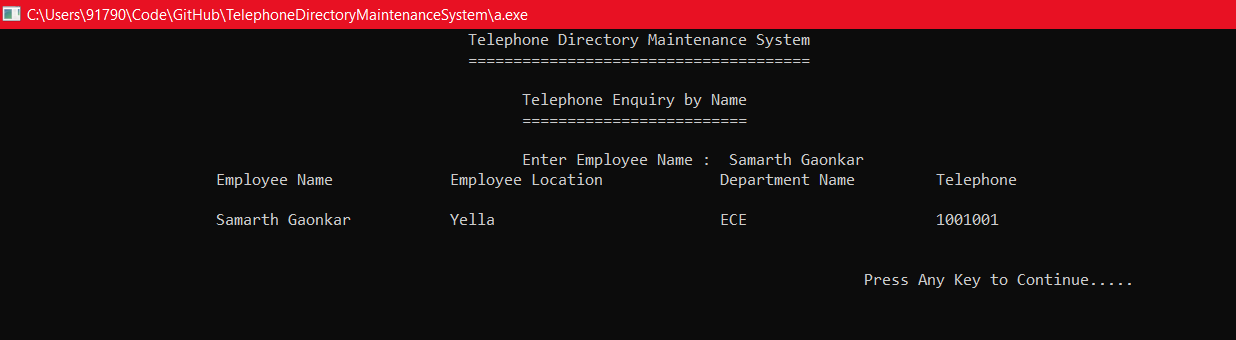
For example :- for I/P “1” Enquiry based on employee name.

For I/P “2” enquiry based on phone numbers.

For I/P “3” Goto main menu of telephone number .

## 5.2.1Telephone Enquiry Based on Name.

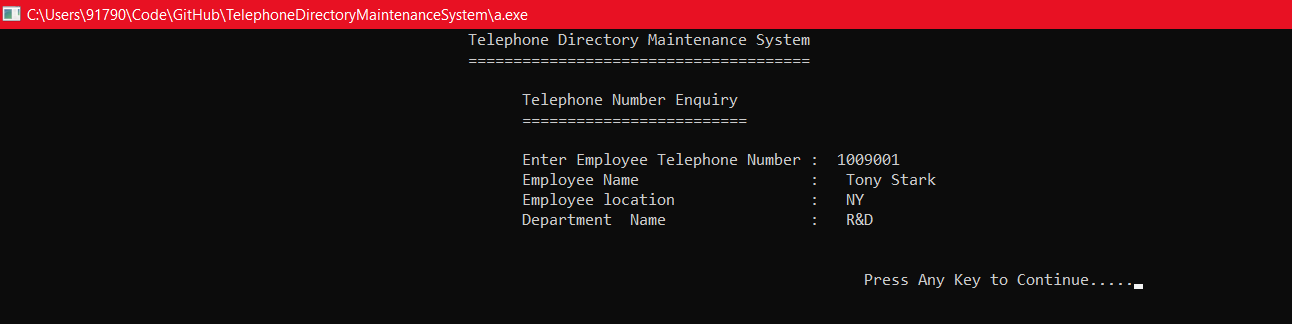
|  |  |
| --- | --- |
|  | **Printing Tell Enquiry based on user input of Name hone number** |



After taking name as input its phone no details is shown.

## 5.2.2Telephone Enquiry Based on Telephone Number

|  |  |
| --- | --- |
|  | **Printing Tell Enquiry based on user input of Phone number** |



After getting user input of phone number This module print all details about Employee.

Source Code Of the Programme

# FileFunction.c

//Telephone directory Maintenance System Project

//project by Samarth Gaonkar USN 2SD17CS080

// Shetty Ganeshprasad USN 2SD17CS088

// Nikhil Vernekar USN 2SD17CS053

// Niranjan Kumar USN 2SD17CS054

//pre processing directory

#include <stdio.h>

#include <string.h>

void system(char \*);

void sleep(int );

int main(void) //start of main function

{

while(1){

int userID, choice; //variables for user id and switch

while (1) //while loop for menu generation

{

system("cls"); //system call for clearing screen

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Login Screen\n ============\n\n");

printf(" Enter the user ID : ");

scanf("%d", &userID); //taking input from user

//Belove code will authenticates the user

if(userID==1780)

{

printf("\n\n\n\n Welcome... Samarth Gaonkar...");

sleep(3);

break;

}

else if(userID==1788)

{

printf("\n\n\n\n Welcome... Shetty Ganeshprasad...");

sleep(3);

break;

}

else if(userID==1753)

{

printf("\n\n\n\n Welcome... Nikhil Vernekar...");

sleep(3);

break;

}

else if(userID==1754)

{

printf("\n\n\n\n Welcome... Niranjan Kumar...");

sleep(3);

break;

}

else

{

printf("\n\n\n\n Invalid User ID... Try Again...");

sleep(3);

}

}

while (1) //while loops for menu generation

{

system("cls"); //system call for clearing screen

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Main Menu\n ==========\n\n");

printf(" 1. Department Maintenance\n");

printf(" 2. Employee Maintenance\n");

printf(" 3. Telephone Directory Maintenance\n");

printf(" 4. Exit\n\n");

printf(" Enter Your Choice : "); //menu generation

scanf("%d", &choice); //taking input from users

switch (choice) //switch statement for menu selection

{

case 1:

system("DeptMaint.exe"); //calling DeptMaint.exe for department maintenance

break;

case 2:

system("EmpMaint.exe"); //calling EmpMaint.exe for Employee maintenance

break;

case 3:

system("TelDir.exe"); //calling Teldir.exe for telephone directory maintenance

break;

case 4:

system("cls"); //calling system call for clearing screen

return (0);

} //emd of switch statement

}

}

return (0);

} //end of main function

# 2)DepTMAINT.c

//Telephone directory Maintenance System Project

//project by Samarth Gaonkar USN 2SD17CS080

// Shetty Ganeshprasad USN 2SD17CS088

// Nikhil Vernekar USN 2SD17CS053

// Niranjan Kumar USN 2SD17CS054

//pre-processing directory

#include <stdio.h>

#include "FileFunctions.h" //importing Custom header file containing all file's functions

void system(char \*);

int main(void) //start of main function

{

int choice; //for switch statement

while(1) //to repeat n times

{

system("cls"); //system csll for clearing screen

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Department Maintenance Menu\n ===========================\n\n");

printf(" 1. Add Department\n");

printf(" 2. Print Department Details\n");

printf(" 3. Goto Main Menu\n\n");

printf(" Enter Your Choice : "); //displaying options for end users

scanf("%d", &choice); //taking input from users

switch (choice) //start of switch statement

{

case 1:

system("cls"); //sysrem call for clearing screen

addDept(); //calling user defined function from Filefunction.h for adding department

break;

case 2:

printDept(); //calling user defined function from Filefunction.h for printing department info

break;

case 3:

return (0); //returning to FileFunction.c

}

} //end of switch statement

} //end of main

# 3)eMPMAINT.c

//Telephone directory Maintenance System Project

//project by Samarth Gaonkar USN 2SD17CS080

// Shetty Ganeshprasad USN 2SD17CS088

// Nikhil Vernekar USN 2SD17CS053

// Niranjan Kumar USN 2SD17CS054

//importing pre-processing directive

#include <stdio.h>

#include "FileFunctions.h" //importing Custom header file containing all file's functions

void system(char \*);

int main(void) //statr of main fubction

{

int choice; //variable for switch

while(1) // to repeat n times

{

system("cls"); //system call for clearing screen

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Employee Maintenance Menu\n ===========================\n\n");

printf(" 1. Add Employee\n");

printf(" 2. Print Employee Details\n");

printf(" 3. Goto Main Menu\n\n");

printf(" Enter Your Choice : "); //displaying options for end users

scanf("%d", &choice); //taking input from users

switch (choice) //start of switch statement

{

case 1:

system("cls"); //system defined function for clearing screen

addEmp(); //calling user defined function from Filefunction.h for adding employee in file

break;

case 2:

printEmp(); //calling user defined function from Filefunction.h for printing employee details from file

break;

case 3:

return (0); //returning to previous function

}

} //end of switch statement

} //end of main function

# 4)TELDIR.c

//Telephone directory Maintenance System Project

//project by Samarth Gaonkar USN 2SD17CS080

// Shetty Ganeshprasad USN 2SD17CS088

// Nikhil Vernekar USN 2SD17CS053

// Niranjan Kumar USN 2SD17CS054

//pre\_processing directive

#include <stdio.h>

#include "FileFunctions.h" //importing Custom header file containing all file's functions

void system(char \*);

int main(void) //start of main function

{

int choice; //variable for switch statement

while(1)

{

system("cls"); //calling system call for clearing screen

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Telephone Directory Maintenance Menu\n ====================================\n\n");

printf(" 1. Add Telephone Number\n");

printf(" 2. Telephone Number Enquiry\n");

printf(" 3. Goto Main Menu\n\n");

printf(" Enter Your Choice : "); //menu genration

scanf("%d", &choice); //taking input from users

switch (choice)

{

case 1:

system("cls"); //system csll for clearing screen

addTele(); //calling user defined function from Filefunction.h for adding telephone info

break;

case 2: while(1)

{

system("cls"); //system call for clearing screen

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Telephone Enquiry Menu\n ======================\n\n");

printf(" 1. Enquiry on Employee Name\n");

printf(" 2. Enquiry on Telephone Number \n");

printf(" 3. Goto Main Menu\n\n");

printf(" Enter Your Choice : "); //menu genertaion

scanf("%d", &choice); //taking input from user

switch (choice)

{

case 1:

enquiryEmp(); //calling user defined function from Filefunction.h for enquiring about employee

break;

case 2:

enquiryTele(); //calling user defined function from Filefunction.h for enquiriing about telephone number

break;

case 3:

break; //return to FileFunction.c

} //end of inner switch statement

if(choice==3)

break;

}

case 3:

return (0);

} //end of outerswitch statement

}//end of while

} //end of main

# 5)fILEfUNCTIONS.H

//Telephone directory Maintenance System Project

//project by Samarth Gaonkar USN 2SD17CS080

// Shetty Ganeshprasad USN 2SD17CS088

// Nikhil Vernekar USN 2SD17CS053

// Niranjan Kumar USN 2SD17CS054

//pre-processing Directory

#include <stdio.h>

#include <string.h>

void system(char \*);

void sleep(int);

//structure for Dept

struct Dept

{

char name[15]; //for storing name

int code; //for storing dept code

};

typedef struct Dept DEPT;

//structure for employee

struct Emp

{

char ename[25]; //for storing employee name

int ecode; //for storing employe code

char dname[15]; //for storing department name

int dcode; //for storing department code

char loc[5]; //for storing location

int tele; //for storing telephone no

};

typedef struct Emp EMP;

//user defined function for adding department

int addDept(void);

int addDept(void)

{

int end;

DEPT d1, d2;

FILE \*fp1; //file pointer for file operation

FILE \*fp2; //file pointer for file operation

char name[15]; //char for stioring name

char dummy;

fp1 = fopen("dept.txt", "a+"); //opening file dept.txt for file operation in append read and write mode

fp2 = fp1;

dummy = getchar();

do

{

fseek(fp2, 0, SEEK\_END); //moving file pointer to end of file

end = ftell(fp2); //storing end location to variable end

system("cls");

if (end == 0)

{

d1.code = 1000;

}

else

{

fseek(fp2, -20, SEEK\_END);

fread(&d2, sizeof(DEPT), 1, fp2);

fseek(fp2, 0, SEEK\_END);

d1.code = (d2.code) + 1;

}

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Add a Department \n");

printf(" Department Code :\t%d\n", d1.code);

printf(" Enter the Department Name :\t");

//scanf("%s", d1.name);

gets(d1.name);

if(strlen(d1.name) > 15)

{

printf(" The Department name is too Big \n");

sleep(3);

}

}while (strlen(d1.name) > 15);

fseek(fp2, 0, SEEK\_SET);

while (fread(&d2, sizeof(DEPT), 1, fp1))

{

strcpy(name, d1.name);

strlwr(name);

strlwr(d2.name);

if (!strcmp(name, d2.name))

{

printf(" The Department name is already exist !!!!\n");

printf(" Press Enter Key To Continue .....");

dummy = getchar();

dummy = getchar();

return (0);

}

}

fwrite(&d1, sizeof(DEPT), 1, fp1);

printf("\n\n Department Added Sucessfully !!!!\n");

printf(" Press Enter Key To Continue .....");

dummy = getchar();

//dummy = getchar();

fclose(fp1); //closeing opened file

}

//user defined function for printing department details

void printDept(void)

{

FILE \*fp1; //file pointer for file opertaion

FILE \*fp2; //file pointer for file operation

DEPT d1;

char dummy;

fp1 = fopen("dept.txt", "a+"); //opening file dept.txt for file operation in append read and write mode

fp2 = fopen("dept.txt", "a"); //opening file dept.txt for file operation in append read and write mode

int end, beg;

fseek(fp2, 0, SEEK\_END);

end = ftell(fp2);

system("cls"); //system csll for clearing screen

printf("\n\n\n\n\n\n\n");

fclose(fp2);

beg = ftell(fp1);

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

if (end == 0)

printf("\t\t\t\t\t\t\t\t\tThere are no Departments!!!!\n");

else

{

printf("\t\t\t\t\t\t\tDepartment Code\t\t\tDepartment Name\n\n");

while (fread(&d1, sizeof(DEPT), 1, fp1))

{

//fread(&d1,sizeof(DEPT),1,fp1);

printf(" \t\t\t\t \t\t\t%4d\t\t\t\t%-25s\n", d1.code, d1.name);

}

fclose(fp1);

}

printf(" \t\t\t\t\t\t\tPress Enter Key to Continue.....");

dummy = getchar();

dummy = getchar();

}

//user defined function for adding employee

int addEmp(void)

{

int end, dCode, flag = 0;

EMP d1;

EMP d2;

DEPT d3;

FILE \*fp1; //file pointer for file operation

FILE \*fp2; //file pointer for file operation

FILE \*fp3; //file pointer for file operation

fp3 = fopen("dept.txt", "a+"); //opening file dept.txt for file operation in append read and write mode

fseek(fp3, 0, SEEK\_END);

if(ftell(fp3)==0)

{

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf("\n\n\n\n\n\n\n\n There are No departments !!!! Please add and Continue.... \n");

sleep(3);

return(0);

}

fseek(fp3, 0, SEEK\_SET);

char dummy;

dummy=getchar();

fp1 = fopen("emp.txt", "a+"); //opening file emp.txt for file operation in append read and write mode

fp2 = fp1;

do

{

system("cls");

fseek(fp2, 0, SEEK\_END);

end = ftell(fp2);

if (end == 0)

{

d1.ecode = 1000;

}

else

{

//to get the emoloyee code of previous employee

fseek(fp2, -64, SEEK\_END);

fread(&d2, sizeof(EMP), 1, fp2);

fseek(fp2, 0, SEEK\_END);

d1.ecode = (d2.ecode) + 1;

}

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Add a Department \n");

printf(" Employee ID :\t%d\n", d1.ecode);

printf(" Enter the Employee Name :\t");

gets(d1.ename);

//scanf("%s", d1.ename);

if(strlen(d1.ename) > 25)

{

printf(" The Employee name is too Big \n");

sleep(3);

}

}while (strlen(d1.ename) > 25);

printf(" Enter the Employee location :\t");

scanf("%s", d1.loc);

while (strlen(d1.loc) > 5)

{

printf(" The Employee location is too Big \n");

printf(" Enter the Employee location :\t");

scanf("%s", d1.loc);

}

printf("\t\t\t\t\t\t\tDepartment Code\t\t\tDepartment Name\n\n");

while (fread(&d3, sizeof(DEPT), 1, fp3))

{

printf(" \t\t\t\t \t\t\t%4d\t\t\t\t%-25s\n", d3.code, d3.name);

}

printf(" (Select the Employee department code from above table)\n");

while (1)

{

printf(" Enter the Employee department code :\t");

scanf("%d", &dCode);

fseek(fp3, 0, SEEK\_SET);

//the below code will searchs department name

while (fread(&d3, sizeof(DEPT), 1, fp3) && (!flag))

{

if (dCode == d3.code)

{

flag = 1;

strcpy(d1.dname, d3.name);

d1.dcode = d3.code;

}

}

if (flag == 0)

{

printf(" Invalid Department code\n");

}

else

{

break;

}

}

if (end == 0)

{

d1.ecode = 1000;

}

else

{

d1.ecode = (d2.ecode) + 1;

}

char name3[25],loc3[5];

strcpy(name3,d1.ename);

strlwr(name3);

strcpy(loc3,d1.loc);

strlwr(loc3);

fseek(fp2,0,SEEK\_SET);

// the below code will handle duplication of employees

while(fread(&d2,sizeof(EMP),1,fp2))

{

strlwr(d2.ename);

strlwr(d2.loc);

if(!strcmp(name3,d2.ename)&&d1.dcode==d2.dcode&&(!strcmp(d2.loc,loc3)))//comparing all parameters

{

printf(" All the details entered is matching with already exist Employee\n");

printf(" Are you sure want to continue??? press any key for yes and n for no :");

scanf(" %c",&dummy);

if(dummy == 'n'||dummy=='N')

{

printf(" Adding Employee is aborted\n");

printf(" Press Enter Key To Continue .....");

dummy = getchar();

dummy = getchar();

return(0);

}

}

}

fwrite(&d1, sizeof(EMP), 1, fp1);

system("cls");

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Add a Department \n");

printf(" Employee ID : %d\n", d1.ecode);

printf(" Employee Name : %s\n", d1.ename);

printf(" Employee location : %s\n", d1.loc);

printf(" Department Code : %d\n", d1.dcode);

printf(" Department Name : %s\n", d1.dname);

printf("\n\n Employee Added Sucessfully !!!!\n");

printf(" Press Enter Key To Continue .....");

dummy = getchar();

dummy = getchar();

system("cls"); //system csll for clearing screen

fclose(fp1);

}

//user defined function for printing employee details

void printEmp(void)

{

FILE \*fp1; //file pointer for file operation

FILE \*fp2; //file pointer for file operation

EMP d1;

char dummy;

fp1 = fopen("emp.txt", "a+"); //opening file emp.txt for file operation in append read and write mode

fp2 = fopen("emp.txt", "a"); //opening file emp.txt for file operation in append read and write mode

int end;

fseek(fp2, 0, SEEK\_END);

end = ftell(fp2);

system("cls"); //system csll for clearing screen

printf("\n\n\n\n\n\n\n");

fclose(fp2);

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

if (end == 0)

printf("\t\t\t\t\t\t\t\t\tThere are no Employees!!!!\n");

else

{

printf("Employee ID \t\tEmployee Name\t\t Employee Location\t\tDepartment code\t\tDepartment Name\t\tTelephone\n\n");

while (fread(&d1, sizeof(EMP), 1, fp1))

{

printf("%4d\t\t\t%-25s %-5s\t\t\t\t%4d\t\t\t%-15s\t\t", d1.ecode, d1.ename, d1.loc, d1.dcode, d1.dname);

if ((d1.tele / 1000) == d1.dcode)

{

printf("%d\n", d1.tele);

}

else

{

printf("\n");

}

}

fclose(fp1);

}

printf("\n\n\t\t\t\t\t\t\t\t\t\t\t\tPress Any Key to Continue.....");

dummy = getchar();

dummy = getchar();

}

//user defined function for adding telephone details

void addTele(void)

{

FILE \*fpe; //file pointer for file operation

FILE \*fpd; //file pointer for file operation

EMP e, temp;

int eCode, flag = 0, count;

int dummy, teleno, i, j;

fpe = fopen("emp.txt", "a+");

fseek(fpe, 0, SEEK\_END);

if(ftell(fpe)==0)

{

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf("\n\n\n\n\n\n\n\n There are No Employees !!!! Please add and Continue.... \n");

sleep(3);

return;

}

fseek(fpe, 0, SEEK\_END);

count = ftell(fpe) / 64;

EMP e1[count];

fpd = fopen("emp.txt", "r");

fread(&e1, sizeof(EMP), count, fpd);

fwrite(&e1, sizeof(EMP), count, fpd);

fclose(fpd);

while (1)

{

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Add Telephone Number \n ====================\n\n");

printf(" Enter the Employee ID :\t");

scanf("%d", &eCode);

for (i = 0; i < count; i++)

{

if (eCode == e1[i].ecode)

{

flag = 1;

break;

}

}

if (flag == 0)

{

system("cls"); //system csll for clearing screen

printf("\n\n\n\n\n\n\n\n\n\n");

printf(" Invalid Employee ID !!!!\n");

printf("\n\n\t\t\t\t\t\t\t\t\t\t\t\tPress Any Key to Continue.....");

dummy = getchar();

dummy = getchar();

system("cls"); //system csll for clearing screen

}

else

{

break;

}

}

teleno = (e1[i].dcode) \* 1000 + 1;

for (j = 0; j < count; j++)

{

if (teleno == e1[j].tele && e1[i].ecode == e1[j].ecode)

{

printf(" Employee Telephone is Already allocated !!!!\n");

break;

}

else if (teleno == e1[j].tele)

{

teleno++;

}

else

{

continue;

}

}

e1[i].tele = teleno;

system("cls");

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Add Telephone Number \n ====================\n\n");

printf(" Employee ID : %d\n", e1[i].ecode);

printf(" Employee location : %s\n", e1[i].loc);

printf(" Department Code : %d\n", e1[i].dcode);

printf(" Telephone Number Allocated : %d\n", e1[i].tele);

printf("\n\n Employee Telephone Number Added Sucessfully !!!!\n");

printf(" Press Any Key To Continue .....");

fclose(fpe);

fpd = fopen("emp.txt", "w");

fwrite(&e1, sizeof(EMP), count, fpd);

fclose(fpd);

dummy = getchar();

dummy = getchar();

system("cls"); //system csll for clearing screen

}

//user defined function for enquiring about employee

void enquiryEmp(void)

{

int dummy;

char name[15], name1[15], name2[15];

FILE \*fp1;

EMP d1;

system("cls"); //system csll for clearing screen

fp1 = fopen("emp.txt", "a+");

fseek(fp1, 0, SEEK\_SET);

fseek(fp1, 0, SEEK\_END);

if(ftell(fp1)==0)

{

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf("\n\n\n\n\n\n\n\n There are No Employees !!!! Please add and Continue.... \n");

sleep(3);

return;

}

fseek(fp1, 0, SEEK\_SET);

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Telephone Enquiry by Name\n =========================\n\n");

printf(" Enter Employee Name : ");

scanf("%s", name);

strcpy(name1, strlwr(name));

printf("\t\t\tEmployee Name\t\t Employee Location\t\tDepartment Name\t\tTelephone\n\n");

while (fread(&d1, sizeof(EMP), 1, fp1))

{

strcpy(name2, d1.ename);

strlwr(name2);

if (!strcmp(name1, name2))

{

printf("\t\t\t%-25s %-5s\t\t\t\t%-15s\t\t", d1.ename, d1.loc, d1.dname);

if ((d1.tele / 1000) == d1.dcode)

{

printf("%d\n", d1.tele);

}

else

{

printf("\n");

}

}

}

fclose(fp1);

printf("\n\n\t\t\t\t\t\t\t\t\t\t\t\tPress Any Key to Continue.....");

dummy = getchar();

dummy = getchar();

}

//user defined function for enquiry telephone details

void enquiryTele(void)

{

int dummy;

int teleno, count = 0;

FILE \*fp1; //file pointer for file operation

EMP d1;

system("cls"); //system csll for clearing screen

fp1 = fopen("emp.txt", "a+");

fseek(fp1, 0, SEEK\_END);

if(ftell(fp1)==0)

{

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf("\n\n\n\n\n\n\n\n There are No Employees !!!! Please add and Continue.... \n");

sleep(3);

return;

}

fseek(fp1, 0, SEEK\_SET);

printf(" Telephone Directory Maintenance System\n ======================================\n\n");

printf(" Telephone Number Enquiry \n =========================\n\n");

printf(" Enter Employee Telephone Number : ");

scanf("%d", &teleno);

while (fread(&d1, sizeof(EMP), 1, fp1))

{

if ((d1.tele) == teleno)

{

printf(" Employee Name : %s\n", d1.ename);

printf(" Employee location : %s\n", d1.loc);

printf(" Department Name : %s\n", d1.dname);

count++;

}

}

if (count == 0)

{

printf(" Invalid Telephone Number !!!! \n");

}

fclose(fp1);

printf("\n\n\t\t\t\t\t\t\t\t\t\t\t\tPress Any Key to Continue.....");

dummy = getchar();

dummy = getchar();

}