Final Report
(Human Resources Management System)

Course Code: CS110 **Course Title:** Computer Programming

Semester: B. Tech 2^{nd} Sem **Section:** S3

Academic Year: 2019-20 **Course Instructor:** Vaishnavi T

Team Members:

1. Diptesh Banerjee, MT011, 8007165183, bdiptesh99@gmail.com

2. Abhishek Chavan, MT001, 8951977709, chavanabhishek527@gmail.com

3. Stavan Kudche, MT046, 8310257468, stavankudche@gmail.com

4. Ishan Bhattacharya, MT017, 6363661297, ishan23301@gmail.com

1 Abstract

Brief Description:

Human Resource Management system(HRMS) is a project developed for software companies to help them in resource allocation and management. The main aim of the project is to manage and update the employee details in the company along with computerization of recruitment process. The implementation of project forms an effective managing platform in the software companies.

Early systems were narrow in scope, typically focused on a single task, such as improving the payroll process or tracking employees' work hours. Today's systems cover the full spectrum of tasks associated with human resources departments, including tracking and improving process efficiency, managing organizational hierarchy, tracking absence and annual leave, simplifying financial transactions, and providing reports on people data. In short, as the role of human resources departments expanded in complexity, HR technology systems evolved to fit these needs.

Human resources software is used by businesses to combine a number of necessary HR functions, such as storing employee data, managing payroll, recruitment, benefits administration, time and attendance, employee performance management, and tracking competency and training records.

Human Resource Management System software acts as a bridge between HR department and Development department in software companies. HRMS looks after overall technical and non technical work details that are carried out in organizations.

Key Features:

1. Login facility
2. Employee recruitment (Based on eligibility)
3. Delete/Remove employee details
4. Update employee details
5. Search employee details
6. Employee salary details
7. Display list of employees
8. Change password
9. Logout

2 Introduction

Every organization, whether big or small, has challenges to overcome. One such challenge is to manage the most important resource of an organization viz., the people serving in it. Quoting John F. Kennedy, the 35th president of the United States-

"Our progress as a nation can be no swifter than our progress in education. The human mind is our fundamental resource."

The human mind is undoubtedly the greatest of all assets, and a powerful tool to manage this is the 'human resource management system'.

Human resource management (HRM) is the strategic approach to the effective management of people in a company or organization such that they help their business gain a competitive advantage. This is done systematically using a 'human resource management system' (HRMS).

HRMS assists the HR manager or the employer in management of human resource. The employer has to first register to be granted access to the features. A registered employer can login using an username and a password. Logging in provides a list of features for the employer to choose from based on the requirement.

The 'recruit' function ensures selection of the right kind of people required for the job. A candidate needs to fulfil certain criteria listed by the function to be selected for the job. If selected, the employer stores the details of the candidate in a file. These details include a new employee ID, name, phone number, mail address, designation and salary. The candidate is now formally an employee of the company.

The portal gives the employee an option to view one's own details. This allows the employee to verify if the details are up to date. In case of discrepancy of details, the employee can report to the employer.

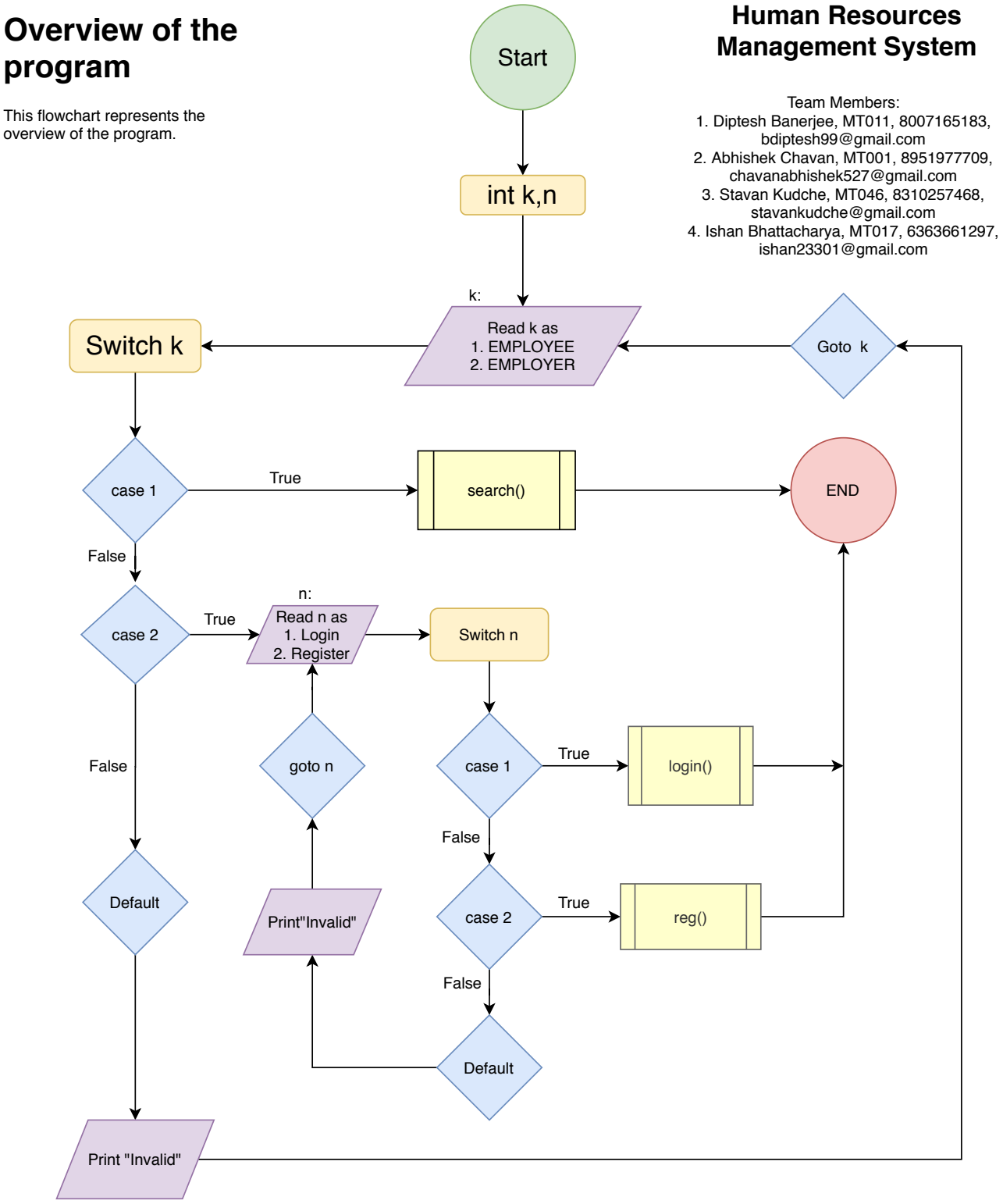
There may arise situations where the details of an employee need to be modified or deleted. In case of the aforesaid circumstances, the employer can access the details using the 'modify' or 'delete' functions to make the corresponding alterations.

The ‘list’ function gives the employer an overview of all employees by displaying all employees’ details. An employer may also search for the details of a particular employee using the ‘search’ function.

In case a registered employer wishes to change one’s login password, the employer may do so after logging in.

Using an HRMS does not need any specific training. It is economical, error-free and saves a lot of time. The data is safe and secure. An HRMS thus simplifies the management of an organization’s most valuable asset and revolutionizes the workplace.

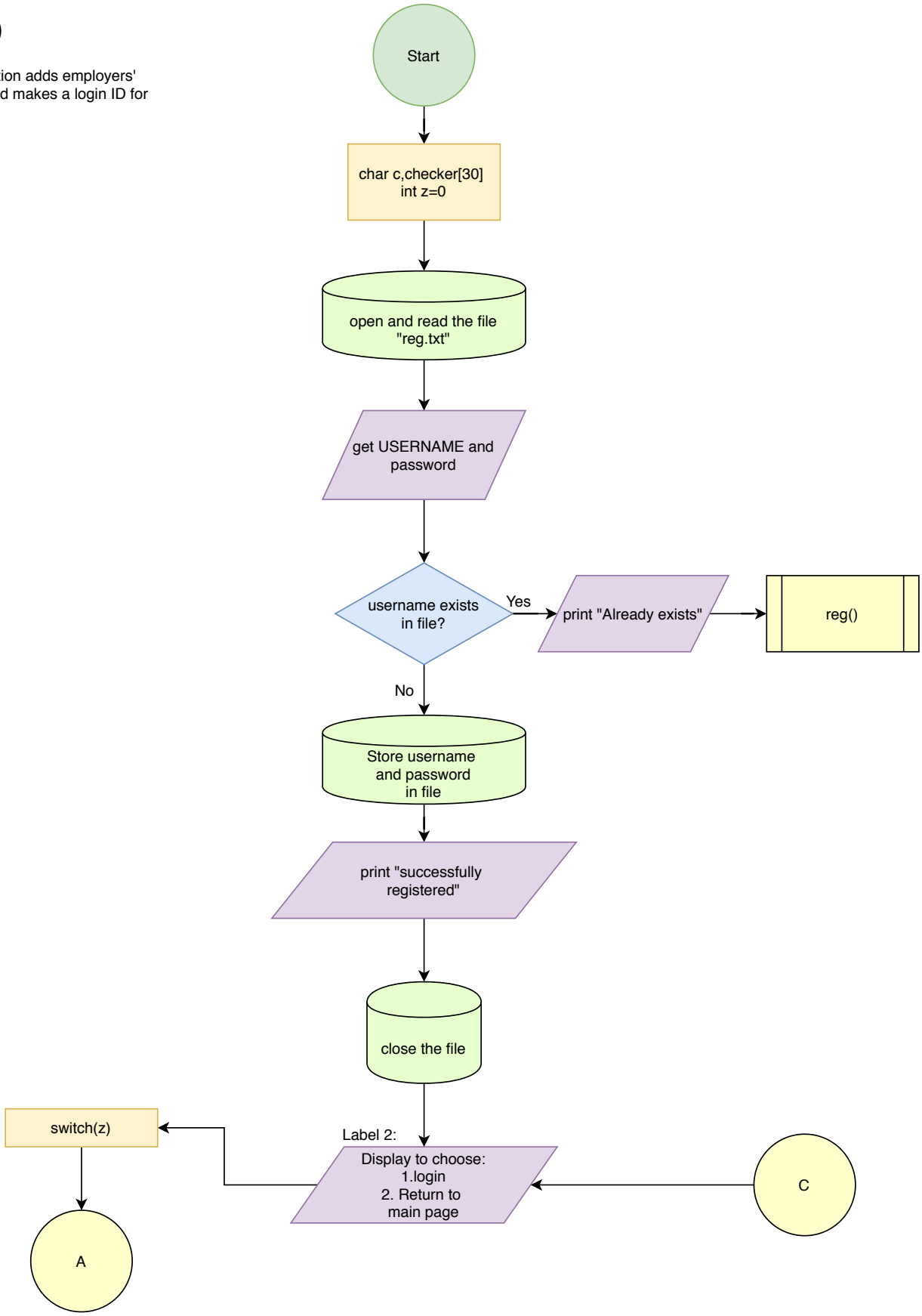
2.1 Flowchart or Algorithm



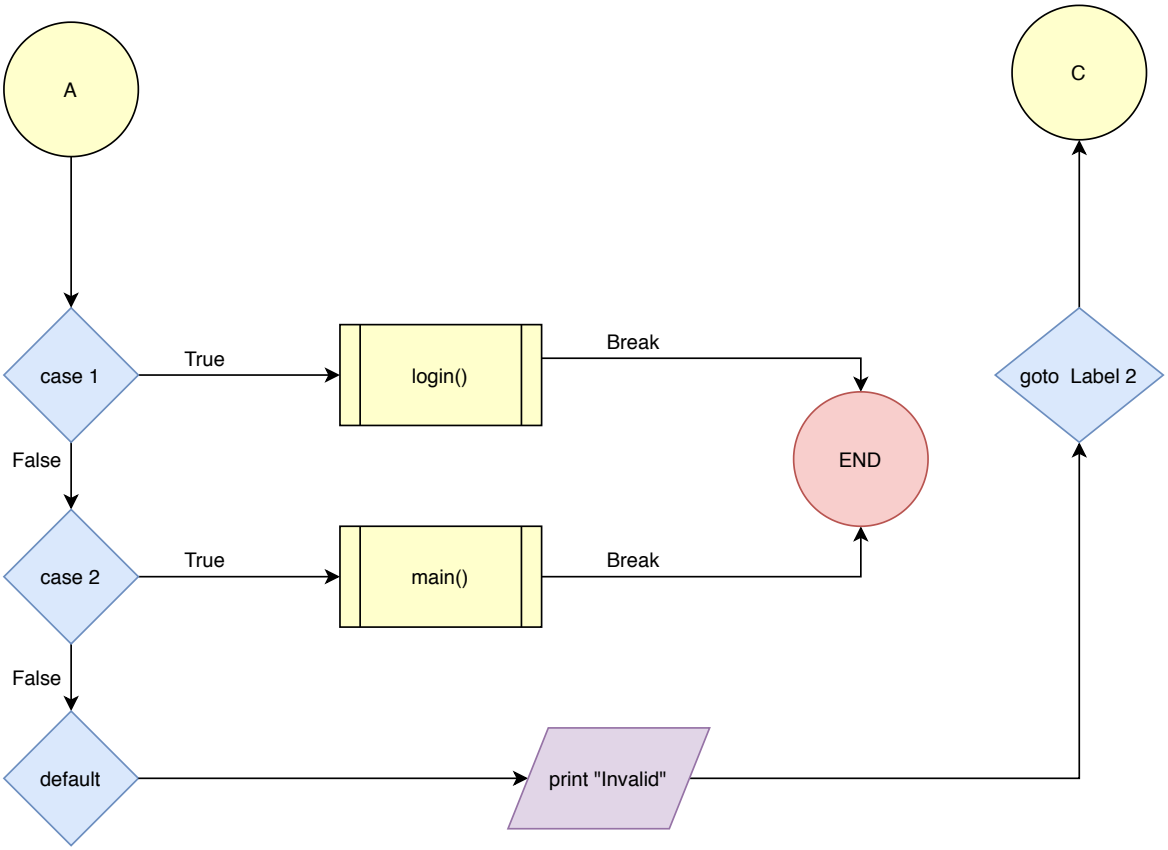
2.2 Flowchart or Algorithm

reg()

This function adds employers' details and makes a login ID for them.



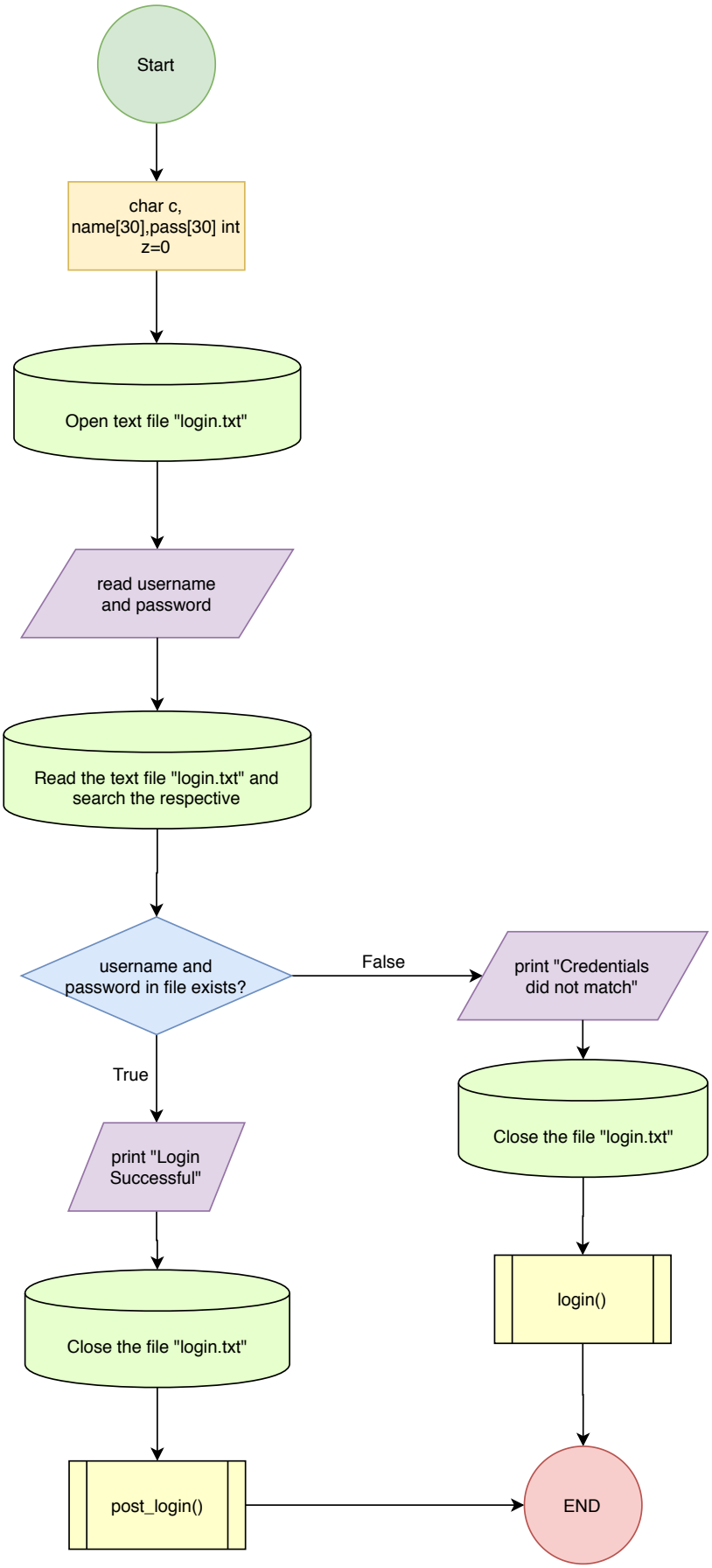
2.3 Flowchart or Algorithm



2.4 Flowchart or Algorithm

login()

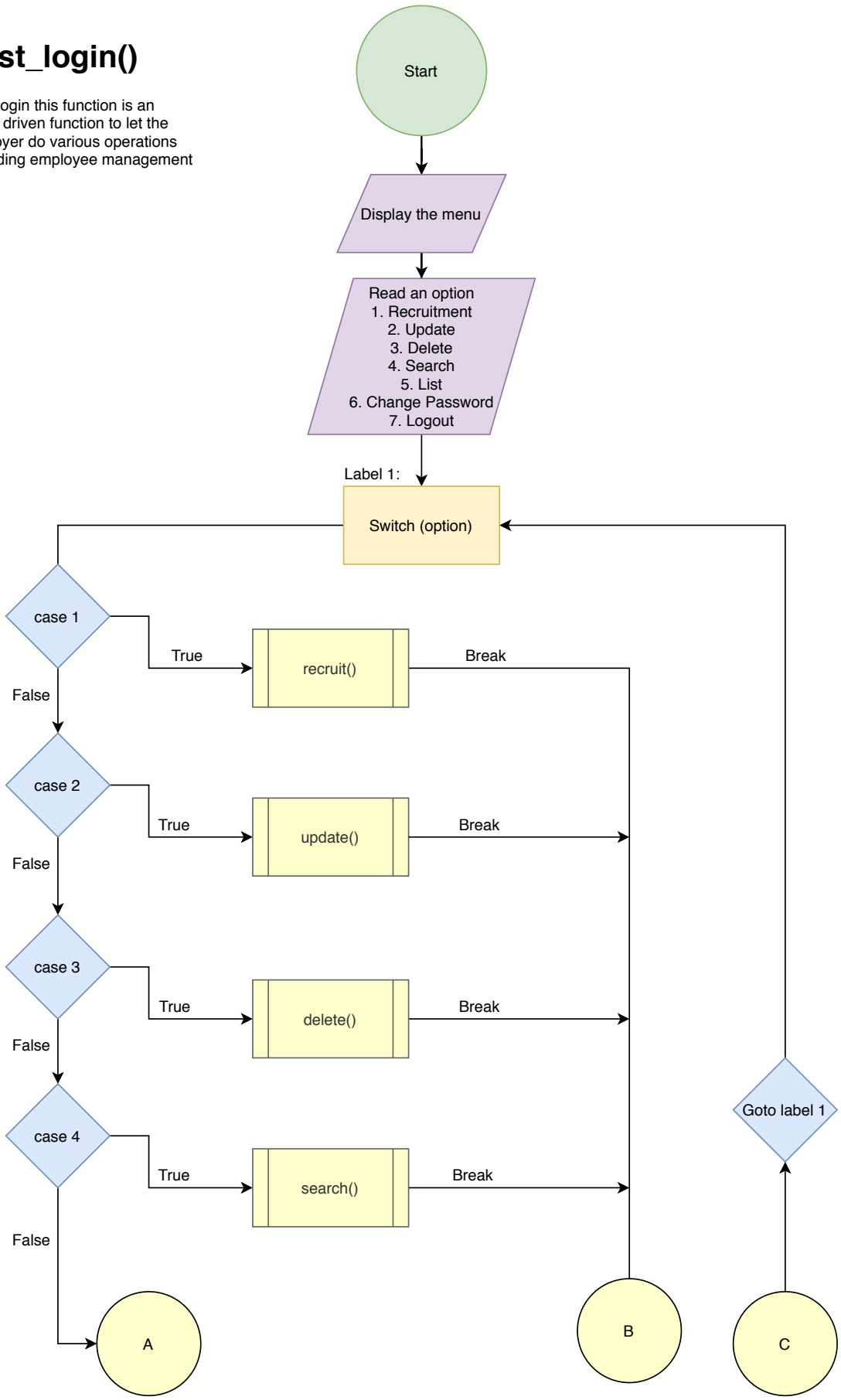
This function lets the employer login if he or she is registered.



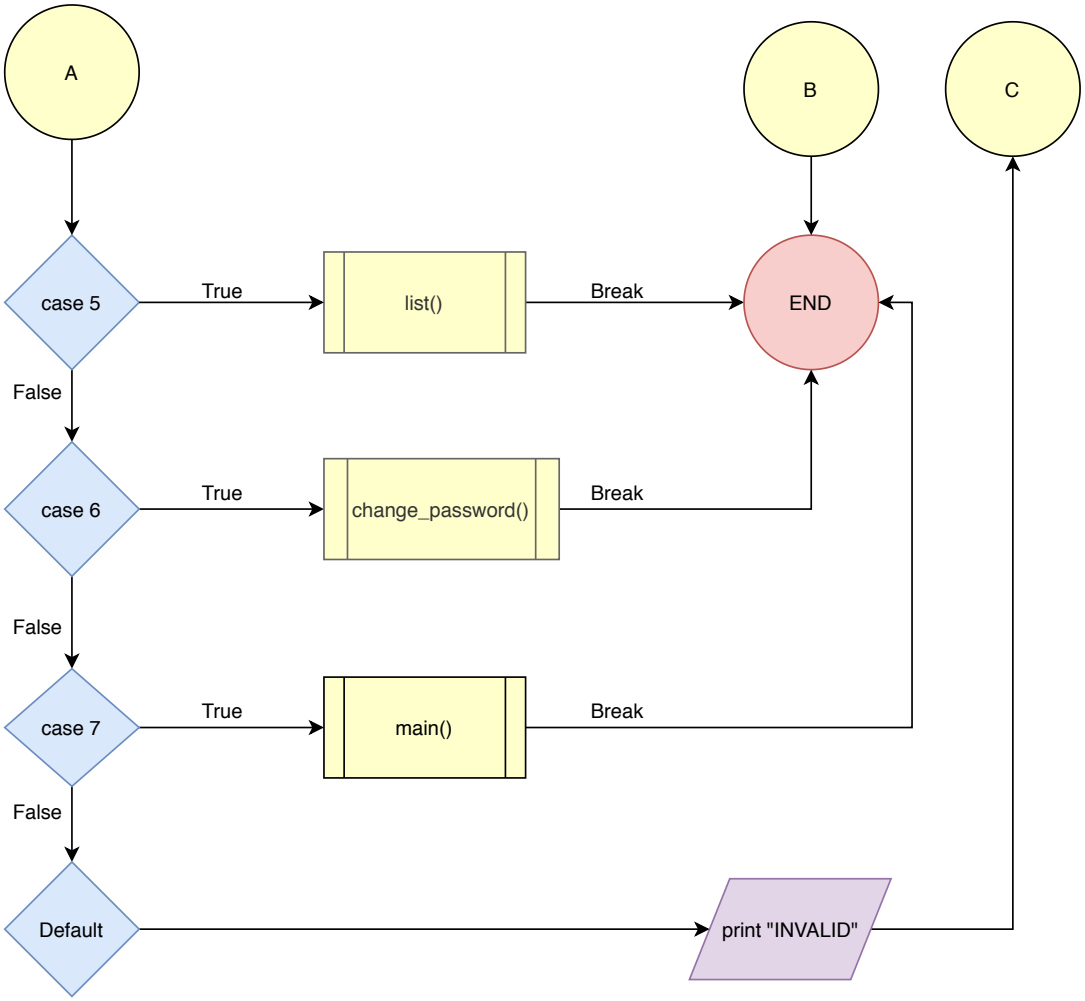
2.5 Flowchart or Algorithm

post_login()

After login this function is an menu driven function to let the employer do various operations regarding employee management



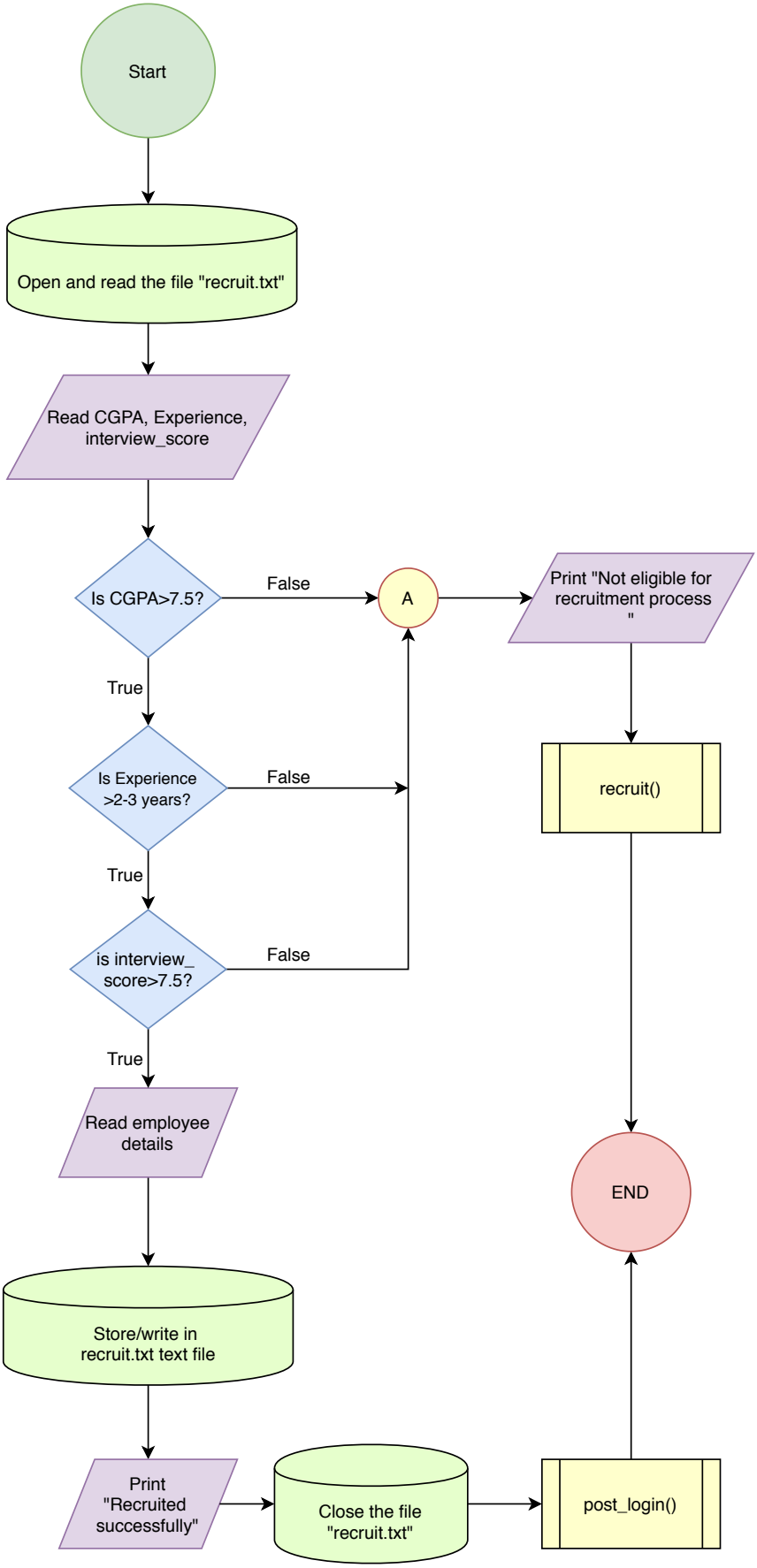
2.6 Flowchart or Algorithm



2.7 Flowchart or Algorithm

recruit()

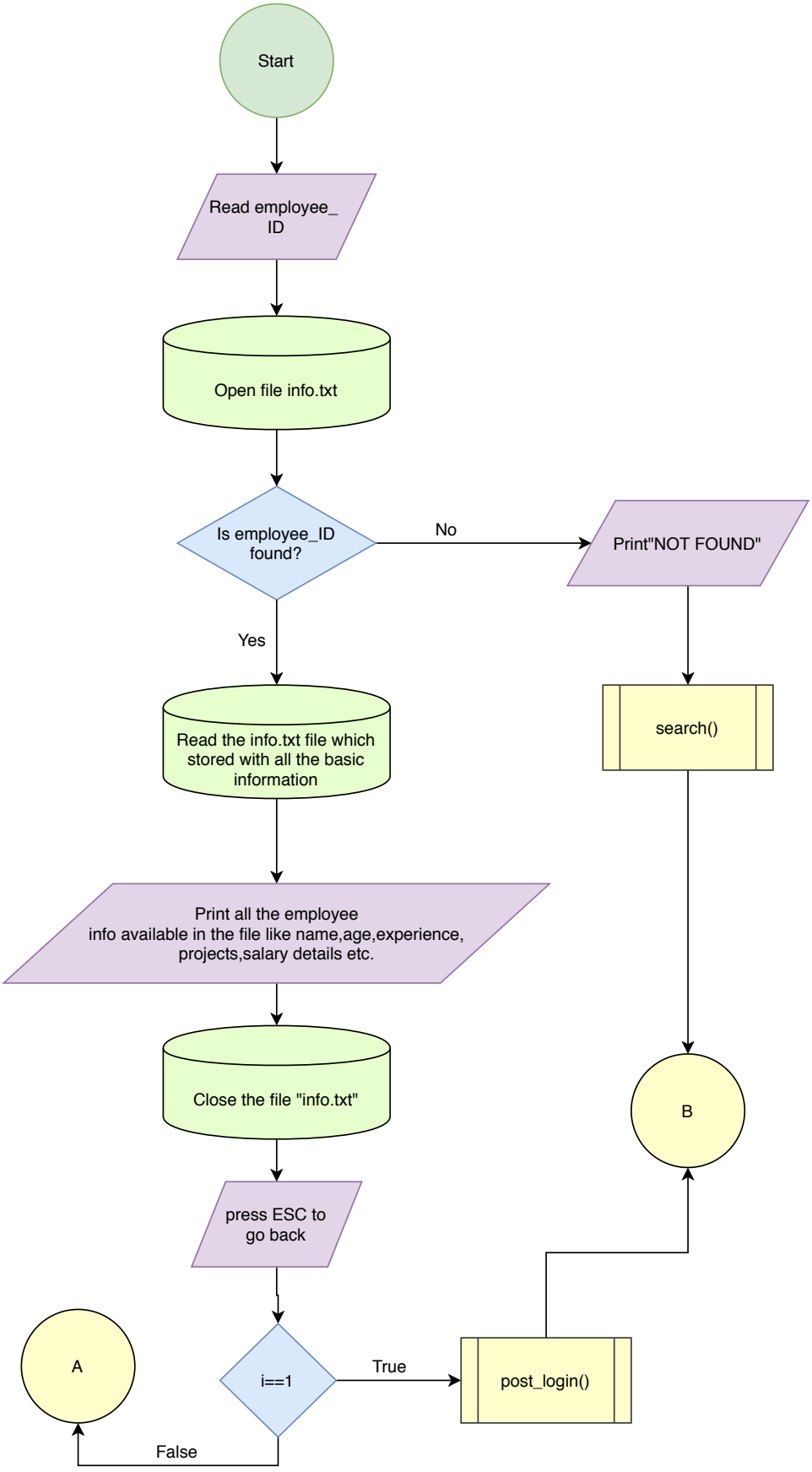
The recruitment process is explained here. It is based on a certain criteria.



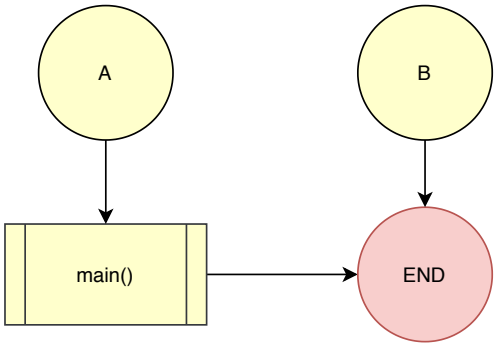
2.8 Flowchart or Algorithm

search(int i)

This is a common function for both employee and employer. It searches the basic details of a particular ID.



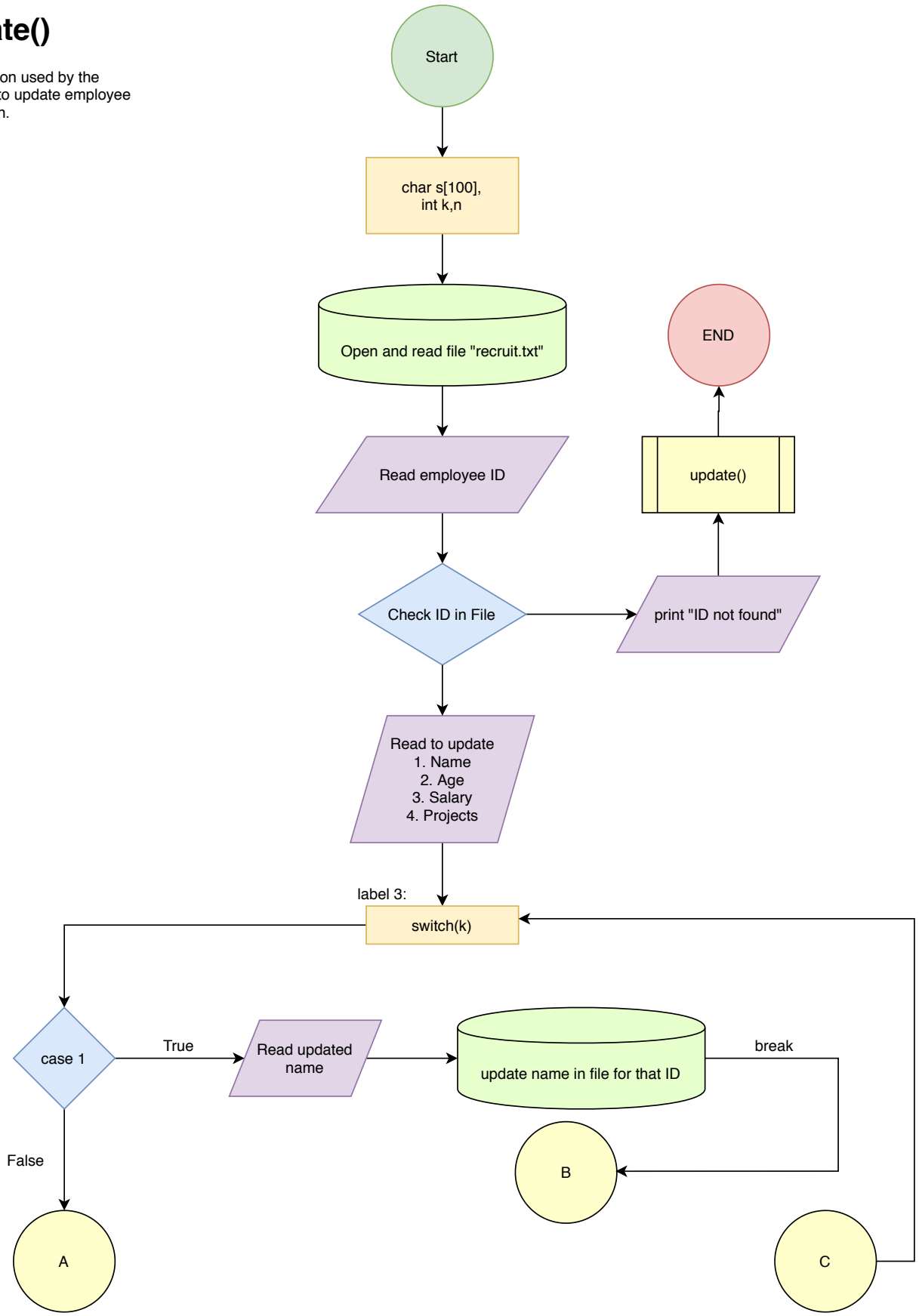
2.9 Flowchart or Algorithm



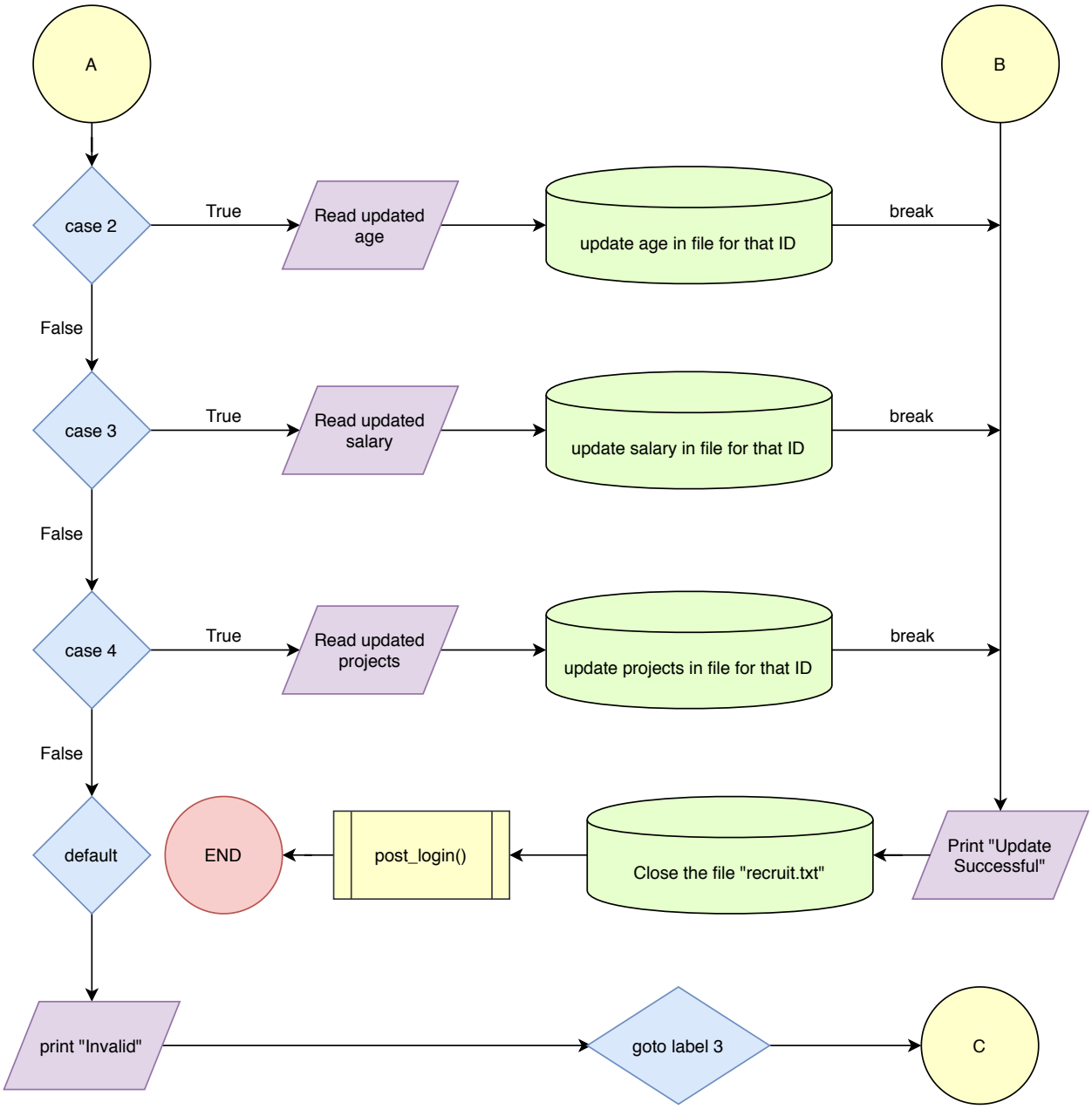
2.10 Flowchart or Algorithm

update()

This function used by the employer to update employee information.



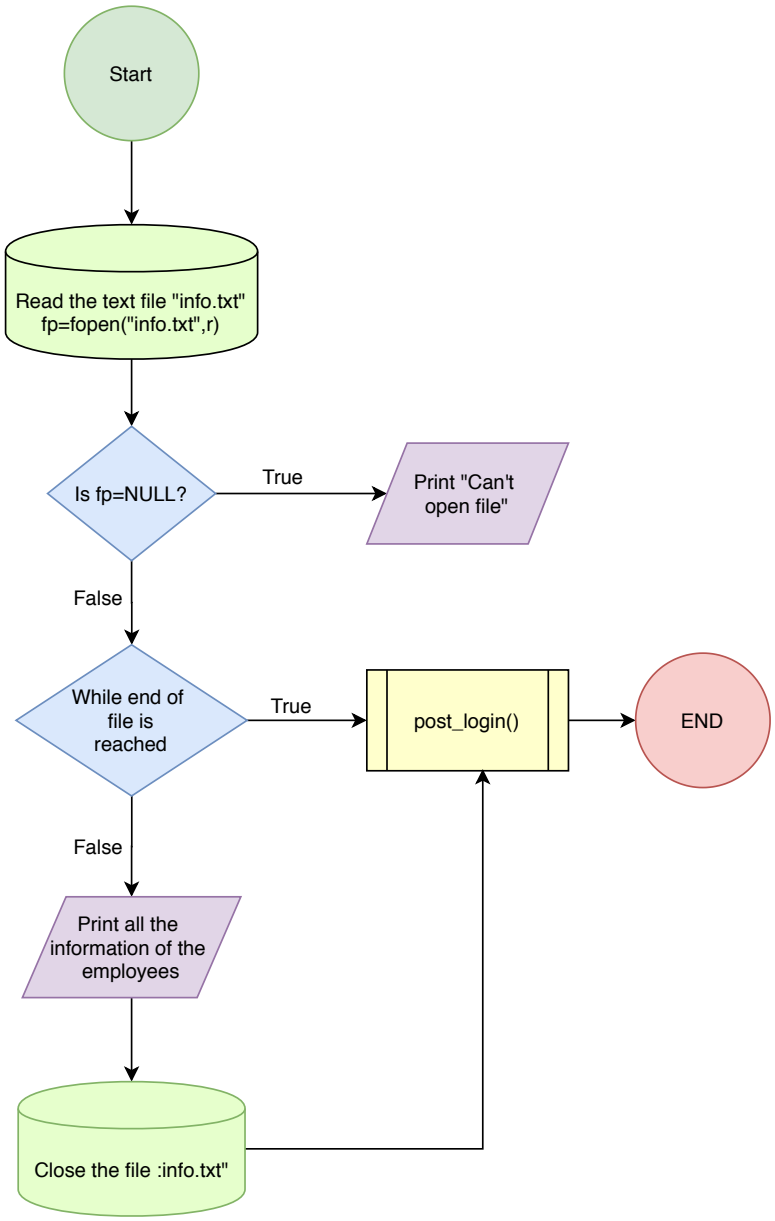
2.11 Flowchart or Algorithm



2.12 Flowchart or Algorithm

list()

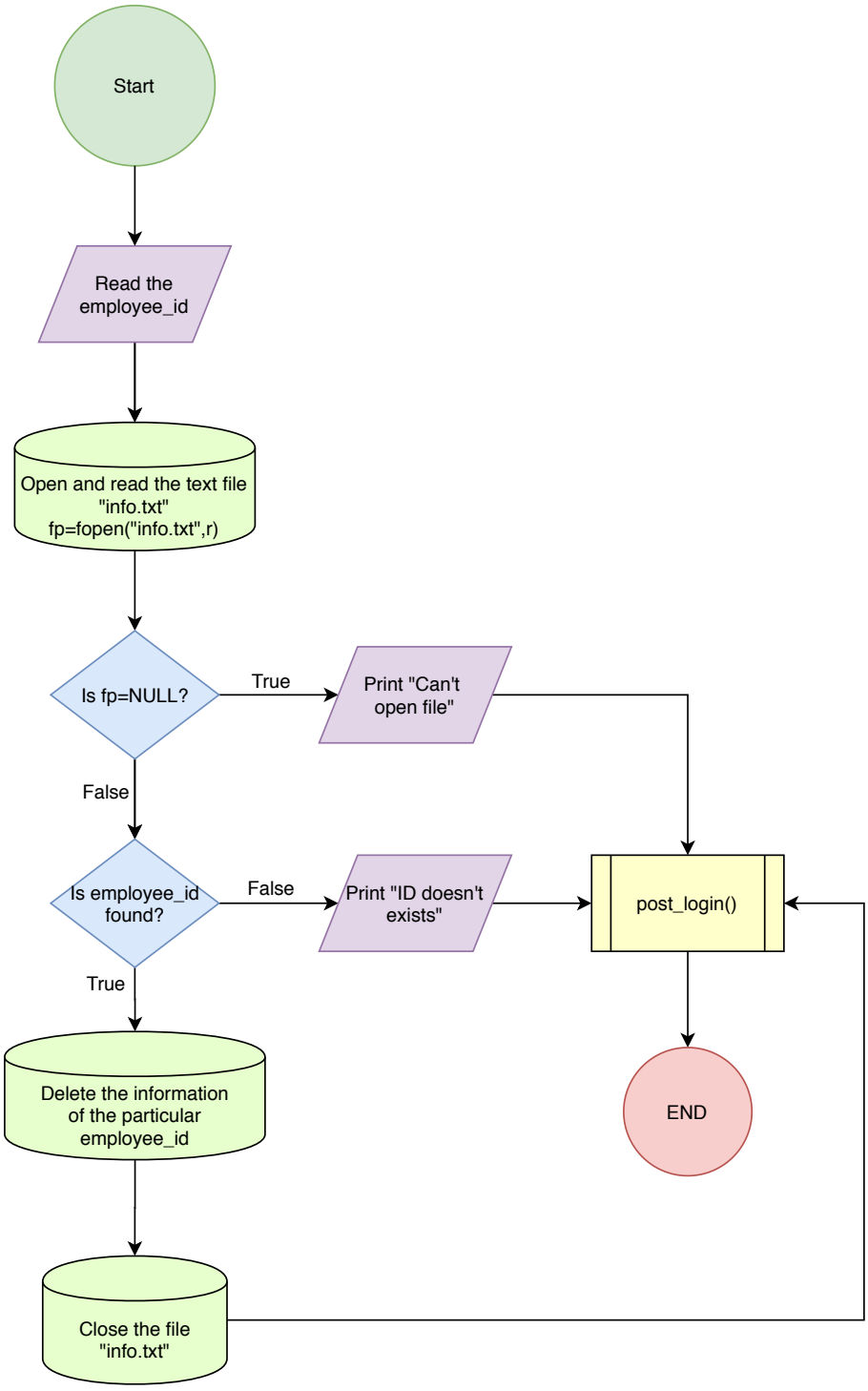
This function is used by the employer to see the basic details of all employees at once.



2.13 Flowchart or Algorithm

delete()

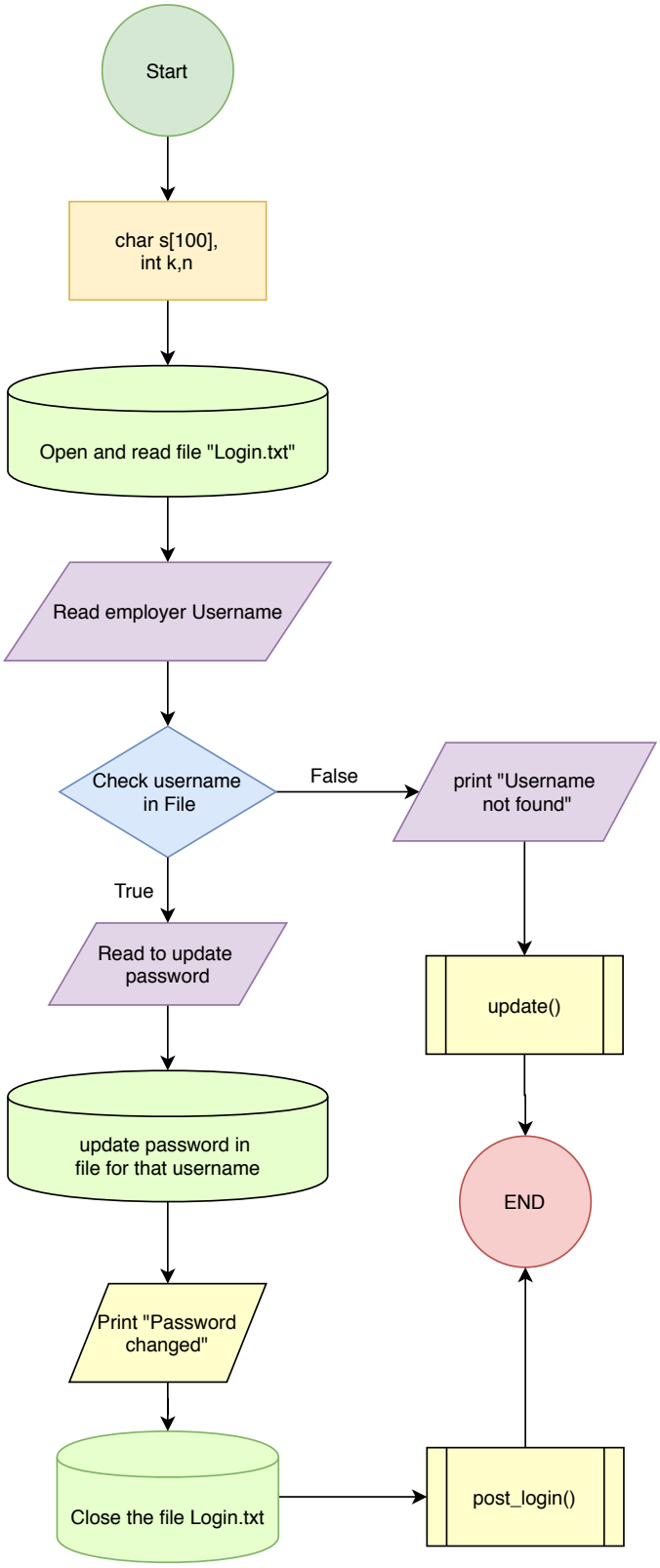
This function is used by the employer to delete data of an employee.



2.14 Flowchart or Algorithm

change_password()

This function lets the employer change his password



3 Source Code

Mini Project HRMS.c

```
//Mini Project by group 16 on
HRMS(Human resource management system).

//Header files to enable the console to work
#include <stdlib.h>
#include<stdio.h>
#include<string.h>

// Function prototypes of various facilities provided.

void login();
void post_login();
void reg();
void recruit();
void search(int i);
void Delete();
void update();
void list();
void change_password();



static int i=0;

//Structures created for basic and detailed information

struct Student{
    char name[30],pass[30];
}w[99];

struct info{
```

```
    char id[10];
    char name[100];
    char no[10];
    char email[100];
    char desig[100];
    char sal[20];
}in;

int n;

void list(){
    FILE *fp;
    fp=fopen("info.txt","r");  //New file info.txt is opened which contains detailed
        information of employee
    struct info inf;
    //Structure variable is declared
    if(fp==NULL){
    printf("\nAn error occurred to open the file!\n");    //Checking whether the file is
        empty
    main(); // calling the main function
    }
    printf("\n\n\t\t
        |==============================================|\
        n");
    printf("\t\t|==============[ LIST OF EMPLOYEES ]==============|\n
        ");
    printf("\t\t
        |==============================================|\
        n");
    int i=0;
    while(fread(&inf, sizeof(struct info), 1, fp)){ //reading the file from first of the
        file info.txt till end
    //Details being printed from the file
```

```
    printf("\n\n\t\t%d.} Name : %s\n\t\t> Employee number : %s\n\t\t> Phone number
        : %s\n\t\t> Designation : %s\n\t\t> Salary(CTC) : %s\n\t\t",i+1,inf.name,inf.id,
        inf.no,inf.desig,inf.sal);
    i++; //updating for new information to be displayed until loop terminates
    }
    fclose(fp);
    printf("\n\n\n\n\n\n\n\n\n\t\t\t\t[ Scroll up..... ]\n\n\n\n\n\n\n\n\n");
    post_login(); //calling function
}

void update(){
    FILE *fp;
    fp=fopen("info.txt","r+");
    if(fp==NULL){
    printf("\nAn error occurred to open the file!\n");
    post_login();
    }


    char empid[10];
    printf("\n\t\t> Update details <");
    printf("\n\n\t\tEnter employee ID : "); //User will enter the ID of which the
        information is to be edited
    scanf(" %[^\n]", empid);
    rewind(fp); //rewind function sets the file position to the beginning of the file
    while(fread(&in,sizeof(in),1,fp)){ // Read all records from file
    if(strcmp(in.id,empid)==0){ // Comparing the entered ID with the one present in the
        file

        // If the condition is true then the information will be updated
        printf("\n\t\t < Enter updated details > ");
        printf("\n\t\tEnter employee name : ");
        scanf(" %[^\n]",in.name);
```

```
        printf("\t\tEnter designation : ");
        scanf(" %[^\n]",in.desig);
        printf("\t\tEnter email : ");
        scanf(" %[^\n]",in.email);
        printf("\t\tEnter contact number : ");
        scanf(" %[^\n]",in.no);
        printf("\t\tEnter salary (CTC) : ");
        scanf(" %[^\n]",in.sal);
        fseek(fp,-sizeof(in),1); // move the cursor one step back from current position
        fwrite(&in,sizeof(in),1,fp); // override the record
        break;
        }
    }

    fclose(fp);
    post_login();
}

void Delete(){
    char another = 'y';
    FILE *fp, *ft;
    fp=fopen("info.txt","r");
    // Checking if file is empty or not
    if(fp==NULL){
     printf("\nAn error occurred to open the file!\n");
            exit(1);
        }

    while(another == 'y'){
        char empname[30];
        printf("\n\t\t> Deleting the details <\n");
        printf("\n\t\tEnter employee ID : "); // Details of the employee details to be
            deleted
```

```
        scanf(" %[^\n] ",empname);
        ft = fopen("temp.txt","w"); // create a intermediate or temporary file for
            temporary storage
        rewind(fp); // sets the file position to the beginning of the file
        while(fread(&in,sizeof(in),1,fp) == 1){ // read all records from file
            if(strcmp(in.id,empname) != 0){ // Checking if the ID exists
            fwrite(&in,sizeof(in),1, ft); // move all records except the one that is to be
                deleted to temp file
            }
        }
        printf("\n\t\t > Deleted successfully! < ");
        fclose(fp);
        fclose(ft);
        remove("info.txt"); // remove the original file
        rename("temp.txt","info.txt"); // rename the temp file to original file name

        printf("\n\t\tWant to Delete another data (y/n)?"); // while loop will be executed
            if required
        scanf("%c",&another);
    }
    post_login();
}

void search(int i){
    FILE *fp;
    char sid[10];
    int k;
    fp=fopen("info.txt","r");
    printf("\n\t\tEnter the employee ID : "); // The ID entered by user to search
    scanf(" %[^\n]",sid);
    while(fread(&in,sizeof(struct info),1,fp)){ // Read the record of file into.txt
        printf("\n\t\t < Details of the employee >\n");
        k=strcmp(sid,in.id); // Comparing whether the ID's match
```

```
        if(k==0){
            // If matched, the details will be displayed
            printf("\n\t\t> Name of employee : %s\n\t\t> Employee ID : %s\n\t\t>
                Contact number : %s\n\t\t> Designation : %s\n\t\t> Salary(CTC) : %s\n\
                t\t",in.name,in.id,in.no,in.desig,in.sal);
            break;
        }
    }
    fclose(fp);
    if(i==1){
        post_login();
    }
    else{
        main();
    }
}

void recruit(){
    FILE *fp1;
    // Declaration of variables
    char another, choice;
    double salary;
    int e_id,n,i;
    char name[100],email[800],post[800],ph[100];
    float cgpa,i_score,xp;

    //Opened file in read mode
    fp1 = fopen("info.txt","rb+");
    if(fp1 == NULL){
        fp1 = fopen("info.txt","wb+"); // Opened file in read mode

        if(fp1 == NULL){
            printf("\nAn error occurred to open the file!\n");
```

```
            exit(1);
        }
    }
    fseek(fp1,0,SEEK_END); // search the file and move cursor to end of the file

    another = 'y';
    while(another == 'y'){ // Loop if user requires to enter multiple recruits
          // Information entered for the new recruits
        printf("\n\t\tEnter the name of the recruit : ");
        scanf(" %[^\n]",name);
        printf("\t\tEnter the CGPA scored in your B.Tech or B.E : ");
        scanf("%f",&cgpa);
        printf("\t\tEnter the experience in the field : ");
        scanf("%f",&xp);
        printf("\t\tEnter the interview score obtained : ");
        scanf("%f",&i_score);

        // Checking the eligibility of the candidate

        if(cgpa>7.5 && xp>2 && i_score>7.5){
            printf("\n\t\t<The candidate has been selected!>\n");
            printf("\n\t\tEnter the new employee ID given : ");
            scanf(" %[^\n]",&in.id);
            printf("\t\tEnter the name of the employee : ");
            scanf(" %[^\n]",in.name);
            printf("\t\tEnter the email address: ");
            scanf(" %[^\n]",in.email);
            printf("\t\tEnter mobile/contact number: ");
            scanf(" %[^\n]",in.no);
            printf("\t\tEnter the designation alloted: ");
            scanf(" %[^\n]",in.desig);
            printf("\t\tSalary annually to be credited: ");
            scanf(" %[^\n]",in.sal);
```

```
            fwrite(&in, sizeof(struct info), 1, fp1); // Writing the details in the file

        }
        else{
            printf("\n\t\t<The candidate is not selected!>\n");
            break;
        }
        printf("\n\t\tWant to add another record?(y/n) : ");
        scanf("\n%c", &another);
    }
    fclose(fp1);
    post_login();
}

void post_login(){
    int i;
    printf("\n\n\t\t=========[ WELCOME TO HRMS ]==========\n");
    label:
        // Main menu to be displayed after login
    printf("\n\t\t <Choose an option>\n\n\t\t 1.Recruitment of new employee\n\t\t 2.
        Update employee detail(s)\n\t\t 3.Delete employee detail(s)\n\t\t 4.Search employee
        detail(s)\n\t\t 5.List of employees\n\t\t 6.Change password\n\t\t 7.Logout\n\t\t"
        );
    printf("\n\t\t Enter your choice : ");
    scanf("%d",&i);
    switch(i)  // Switch case for the option selected
    {
        case 1:  // Below are all the function call for
                recruit();
                break;
        case 2:
                update();
                break;
```

```
        case 3:
                Delete();
                break;
        case 4:
                search(1);
                break;
        case 5:
                list ();
                break;
        case 6:
                change_password();
                break;
        case 7:
                main();
                break;
        default:
            printf("\n\t\tInvalid input! Enter valid input.\n"); // Statement for incorrect
                input
            goto label;
    }
}

void reg(){
    int k;
    int temp=0;
    char check[50];
    FILE *fp;
    fp= fopen ("reg.txt", "a"); // opening file in append mode
    if (fp == NULL) {
        printf("\nAn error occurred to open the file!\n");
        main();
    }
    printf("\n\n\t\tNumber of people to be registered: "); // Entering new registration
```

```
    details
scanf("%d",&k);
for(int i=0;i<k;i++){
    printf("\n\t\tEnter username : ");
    scanf("%s",w[i].name);
    printf("\n\t\tCreate password : ");
    scanf("%s",w[i].pass);
    fwrite (&w[i], sizeof(struct Student), 1, fp);
}
//fwrite (&inp1, sizeof(struct Student), 1, of);
if(fwrite != 0)
    printf("\n\n\t\tRegistered successfully to the system!\n");
else
    printf("\n\n\t\tError writing file !\n");
fclose (fp);
AA: //label
printf("\n\t\t*****************************************");
printf("\n\t\tChoose from the MINI MENU below: ");
printf("\n\t\t*****************************************");
printf("\n\n\t\t 1. Go back to login page\n\n\t\t 2. Home\n");
printf("\n\t\tEnter option : "); // Mini menu for user
scanf("%d",&temp);
switch(temp){
    case 1:
        login();
        break;
    case 2:
        main();
        break;
    default:
        printf("\n\t\tInvalid input! Enter valid input.\n");
        goto AA;
}
```

```
}

void change_password(){
    FILE *fp;
    fp=fopen("reg.txt","r+");
    if(fp==NULL){
        printf("\nAn error occurred to open the file!\n");
        post_login();
    }
    struct Student st;
    char s[10],p[10];
    printf("\n\t\t> Change Password <");
    printf("\n\n\t\tEnter employer username : "); //User will enter the username of which
        the information is to be edited
    scanf(" %[^\n]", s);
    printf("\n\n\t\tEnter Current Password:"); //User Will enter his current password
    scanf(" %[^\n]", p);
    rewind(fp); //rewind function sets the file position to the beginning of the file
    while(fread(&st,sizeof(st),1,fp)){ // Read all records from file
        if(strcmp(st.name,s)==0 && strcmp(st.pass,p)==0){ // Comparing the entered
            username and password with the one present in the file

            // If the condition is true then the information will be updated
            printf("\n\t\t< Enter new details > ");
            printf("\n\t\tEnter employer username : ");
            scanf(" %[^\n]",st.name);
            printf("\t\tEnter New password : ");
            scanf(" %[^\n]",st.pass);
            fseek(fp,-sizeof(st),1); // move the cursor one step back from current position
            fwrite(&st,sizeof(st),1,fp); // override the record
            break;
        }
    }
```

```
    fclose(fp);
    post_login();
}

void login(){
    // Declaration of variables
    char nam[50],pas[20];
    int chkp,chku,temp,q;
    FILE *inf;
    //Creating an object variable
    struct Student inp;
    inf = fopen("reg.txt", "r");
    if (inf == NULL) {
        printf("\nError to open the file\n");
        exit (1);
    }
    // Login details will be entered by users
    printf("\n\t\tEnter Username: ");
    scanf("%s",nam);
    printf("\n\t\tEnter Password: ");
    scanf("%s",pas);
    while(fread(&inp, sizeof(struct Student), 1, inf)){  // Checking if the details are
        available in the file
        chku=strcmp(nam,inp.name);
        chkp=strcmp(pas,inp.pass);
        if(chku==0 && chkp==0){
            temp=0;
            break;
        }
        else{
            temp=1;
        }
```

```
    }
    fclose (inf);
    switch(temp){ // Switch case for login status
        case 0:
            printf("\n\t\t<Login Successful!>\n");
            post_login();
            break;
        case 1:
            printf("\n\t\t<Wrong Credentials!>\n");
            AA: //label
            printf("\n\t\t****************************************");
            printf("\n\t\tChoose from the MINI MENU below: ");
            printf("\n\t\t****************************************");
            printf("\n\t\t 1. Login\n\n\t\t 2. Register\n"); // Mini menu
            printf("\n\t\tEnter option : ");
            scanf("%d",&q);
            switch(q){
                case 1:
                    login();
                    break;
                case 2:
                    reg();
                    break;
                default:
                    printf("\n\t\tInvalid input! Enter valid input.\n");
                    goto AA;
            }
        break;
    }
}

int main(){
    int k;
```

```
// First menu that will be displayed as program is executed
  printf("\n\n\n\n\n\t\t\t\t
      ==============================================="
      );
 printf("\n\t\t\t\t       [ HUMAN RESOURCES MANAGEMENT SYSTEM (HRMS) ]
     ");
 printf("\n\t\t\t\t
     ==============================================")
     ;
YZ: //label
 printf("\n\n\n\n\n\t\t\t\t\t\tAre you an.. ");
 printf("\n\n\n\t\t\t\t1. Employee?\t\t\t2. Employer?");
 printf("\n\n\n\t\t\t\t\t\t> Enter a choice : ");    // Options to be selected by the
     user
scanf("%d",&k);
switch(k){
     case 1:
          search(0);
          break;
     case 2:{
          XY:
          printf("\n\n\n\t\t\t\t1. Login\t\t\t2. Register");
          printf("\n\n\n\t\t\t\t\t\t> Enter a choice : ");
          scanf("%d",&n);
          switch(n){
               case 1:
                    login();
                    break;
               case 2:
                    reg();
                    break;
               default:
                    printf("\n\n\t\t\t\t\t\tNo match found! Enter valid options. ");
```

```
                printf("\n\n\t\t\t\t\tPress \"Enter\" to re-Enter the choice");
                if(getch()==13){
                goto XY;
                }
            }
            return 0;
            break;
        }
        default:
            printf("\n\n\t\t\t\t\tNo match found! Enter valid options. "); // Statement
                for invalid inputs
            printf("\n\n\t\t\t\t\tPress \"Enter\" to re-Enter the choice.");
            if(getch()==13){
            goto YZ;
            }
    }
}
```

4 Results

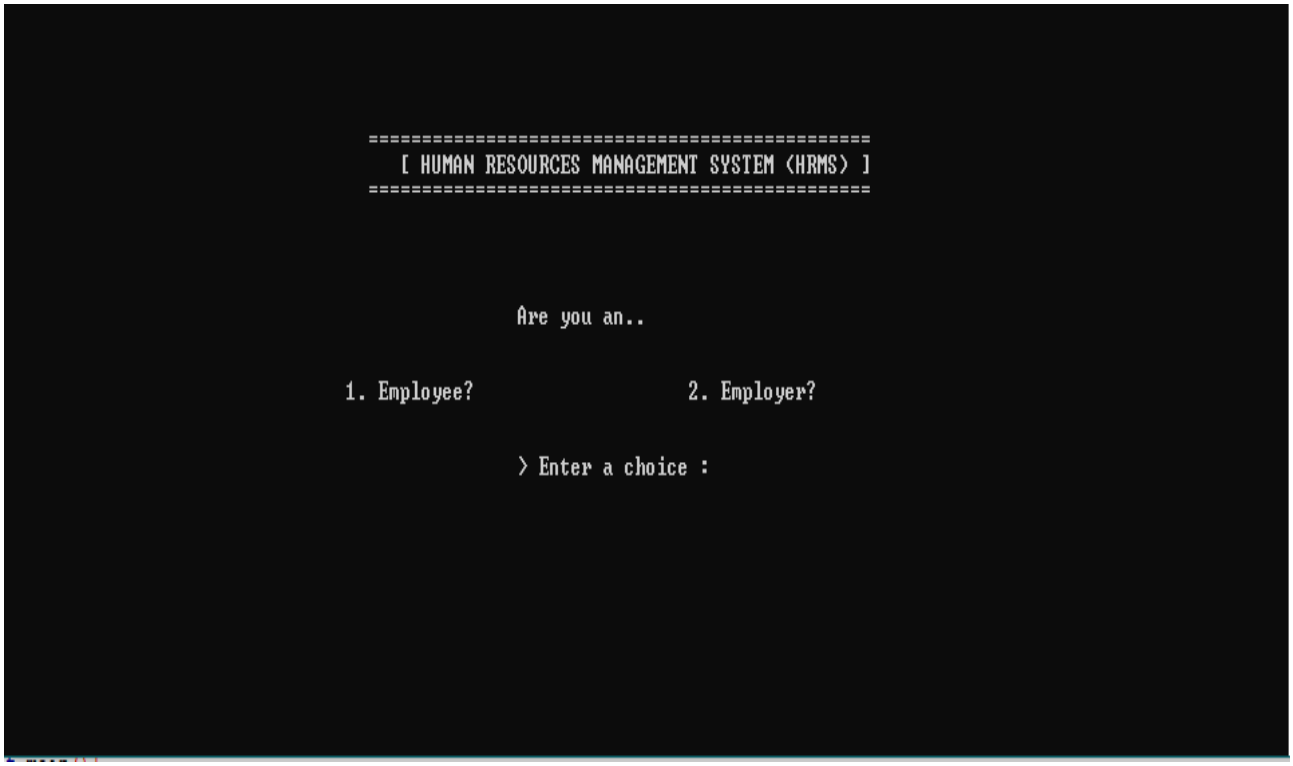


Figure 1: Main Page

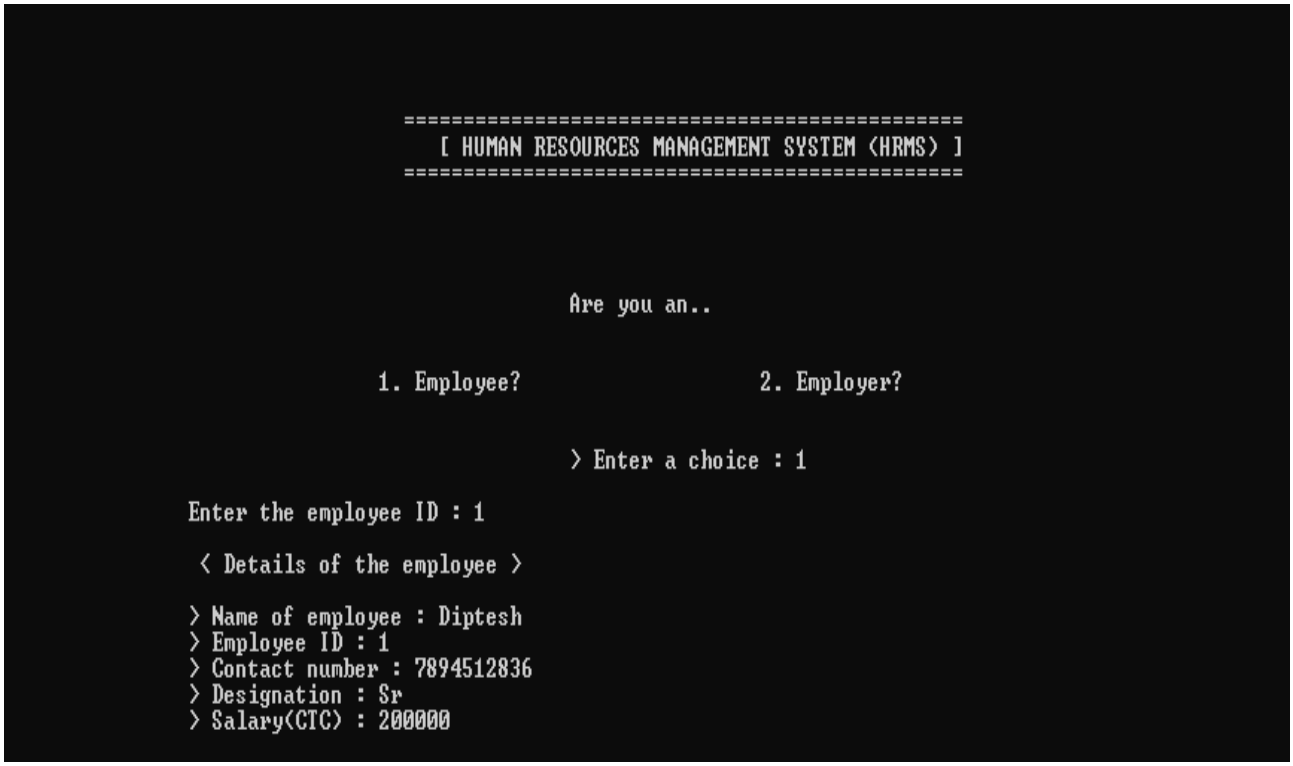


Figure 2: If you select Employee

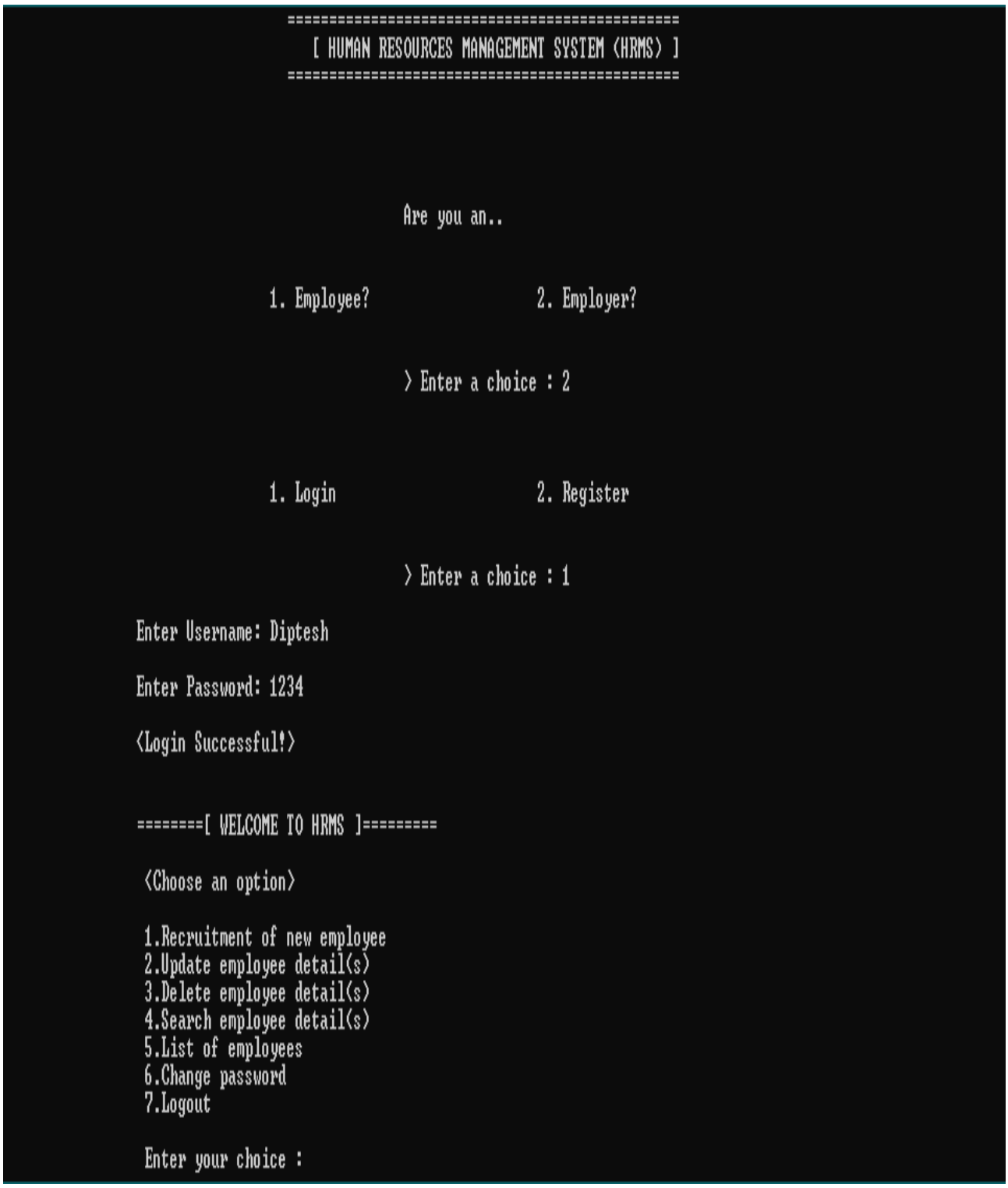


Figure 3: The login

```
========[ WELCOME TO HRMS ]=========

 <Choose an option>

 1.Recruitment of new employee
 2.Update employee detail(s)
 3.Delete employee detail(s)
 4.Search employee detail(s)
 5.List of employees
 6.Change password
 7.Logout

 Enter your choice : 1

Enter the name of the recruit : Aakash
Enter the CGPA scored in your B.Tech or B.E : 9.6
Enter the experience in the field : 8
Enter the interview score obtained : 8.4

<The candidate has been selected!>

Enter the new employee ID given : 2
Enter the name of the employee : Aakash
Enter the email address: aka@gmail.com
Enter mobile/contact number: 7894512360
Enter the designation alloted: Sr. aa
Salary annually to be credited: 500000

Want to add another record?(y/n) : n


========[ WELCOME TO HRMS ]=========

 <Choose an option>

 1.Recruitment of new employee
 2.Update employee detail(s)
 3.Delete employee detail(s)
 4.Search employee detail(s)
 5.List of employees
 6.Change password
 7.Logout

 Enter your choice :
```

Figure 4: The Recruitment Process

```
 <Choose an option>

 1.Recruitment of new employee
 2.Update employee detail(s)
 3.Delete employee detail(s)
 4.Search employee detail(s)
 5.List of employees
 6.Change password
 7.Logout

 Enter your choice : 2

> Update details <

Enter employee ID : 2

 < Enter updated details >
Enter employee name : Aakash
Enter designation : Sr. Sales
Enter email : aka@gmail.com
Enter contact number : 7894512360
Enter salary (CTC) : 600000


========[ WELCOME TO HRMS ]=========

 <Choose an option>

 1.Recruitment of new employee
 2.Update employee detail(s)
 3.Delete employee detail(s)
 4.Search employee detail(s)
 5.List of employees
 6.Change password
 7.Logout

 Enter your choice : 5


|================================================|
|=============[ LIST OF EMPLOYEES ]=============|
|================================================|


1.> Name : Diptesh
> Employee number : 1
> Phone number : 7894512836
> Designation : Sr
> Salary(CTC) : 200000


2.> Name : Aakash
> Employee number : 2
> Phone number : 7894512360
> Designation : Sr. Sales
> Salary(CTC) : 600000
```

Figure 5: Updating employee Information

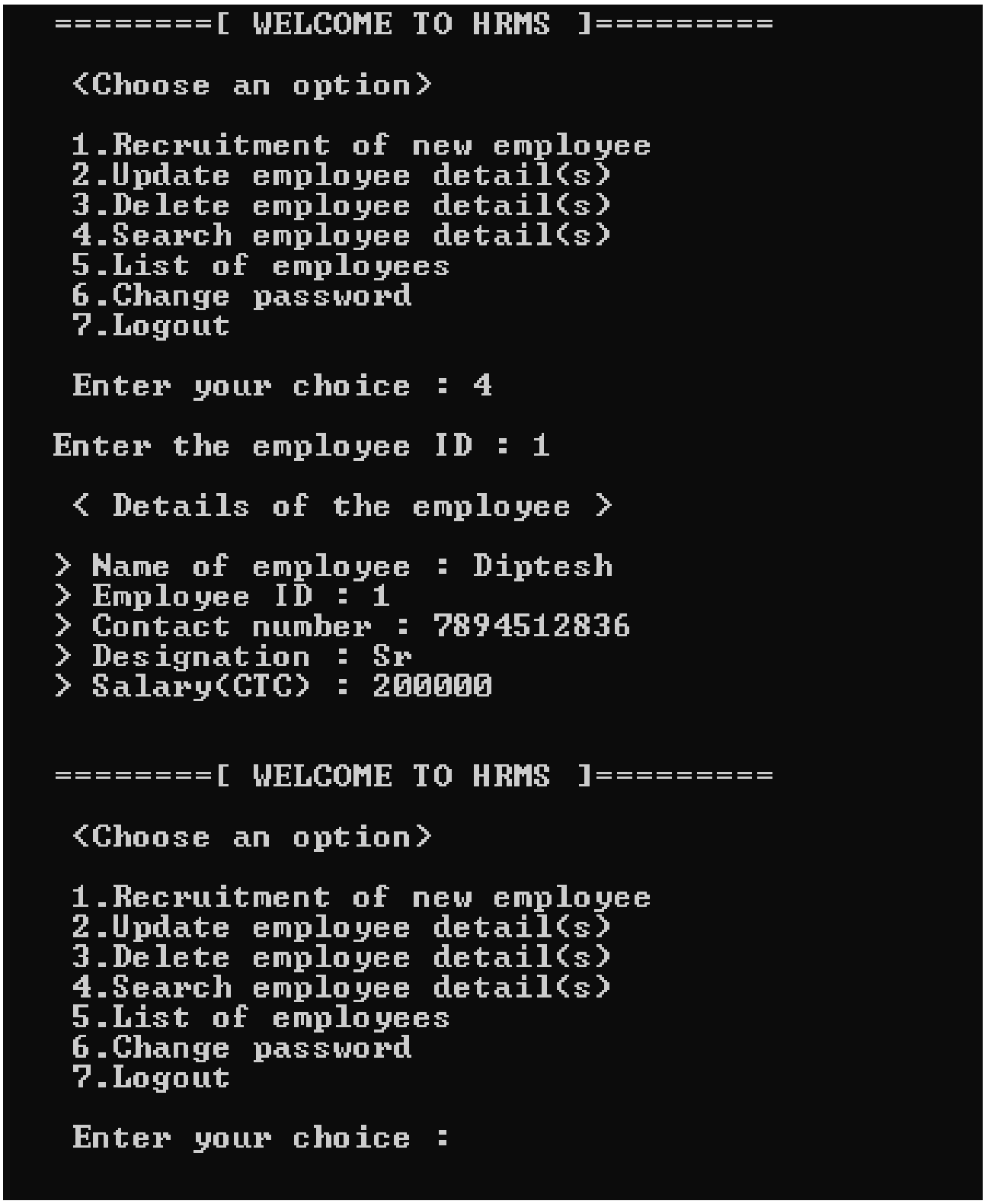


Figure 6: Searching employees

Figure 7: Removing Employees

Figure 8: To change Password of the Employer

Figure 9: Login after changeing password

5 References:

1. http://programmingknowledgeblog.blogspot.com/2013/01/c-mini-project-human-resource.html

2. https://1000projects.org/human-resource-management-system.html

****** END ******