Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 03, March -2018

SALARY BASED MANAGEMENT SYSTEM USING FACE RECOGNITION AND FILE HANDLING

Prabu S, Professor[1], Akarsh Gupta[2]

[1]*Professor.VIT University, Vellore.*
[2]*VIT University, Vellore.*

ABSTRACT-*We are implementing a company salary database management with a basic GUI implemented in C++ that generates salary of employee of a particular department with the ability to generate Salary receipt and all the information regarding the employee. And another feature is that the attendance of the employees will be marked using face recognition.*
We have made use of concepts of database like Normalization, Primary Key and Foreign key concepts. Our Database is implemented on C++.We have also added the salary receipt printing function in it.
"Payroll Management System" is one of the core areas of your business. Usually, it is pursued to manage the employees the employee's expenses, Allowances, salary, Gross Salary, Deduction, Tax and many more for a specific time period. Management and Accounting are two main essential parts for payroll.
Payroll is an area in which you do not want to take any risk because it leads to some financial and serious legal consequences. Payroll is a serious concern for every SME. It is mandatory for all business to pay every employee as per the government rules and regulations.
Furthermore, this project will develop for company management and enhance business in market and maintain the prestigious and reputation of the company. Others, this project to facilitate company to handle all the legal process and employee's expenditure properly and systematically.
For Face Recognition we use a python code which is based on Local Binary Pattern Histogram method.

INTRODUCTION

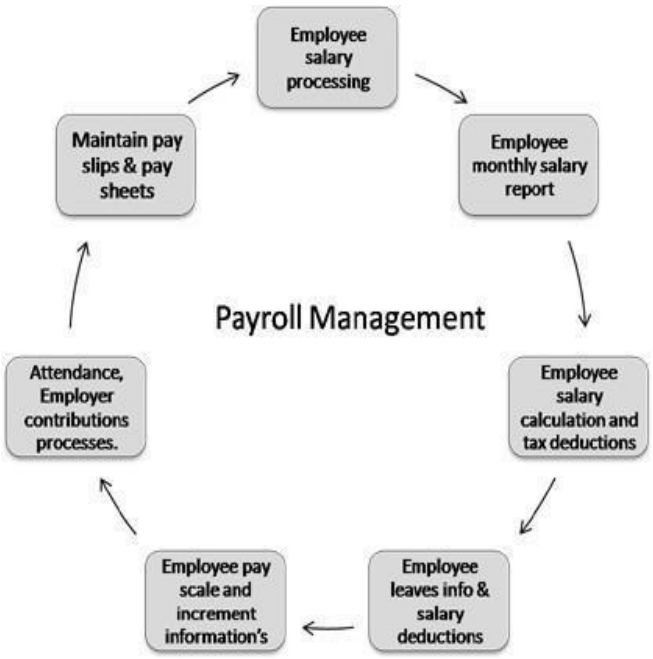
The Salary Database Management System deals with the financial aspects of employee's salary, allowances, deductions, gross pay, net pay etc. and generation of pay-slips for a specific period.

The outstanding benefit of Salary Management System is its easy implementation. Other advantages of Salary Database Management System are its extensive features and reports. And another great benefit will be that it will be easier to record at what time the employee came to work using face recognition.

Features

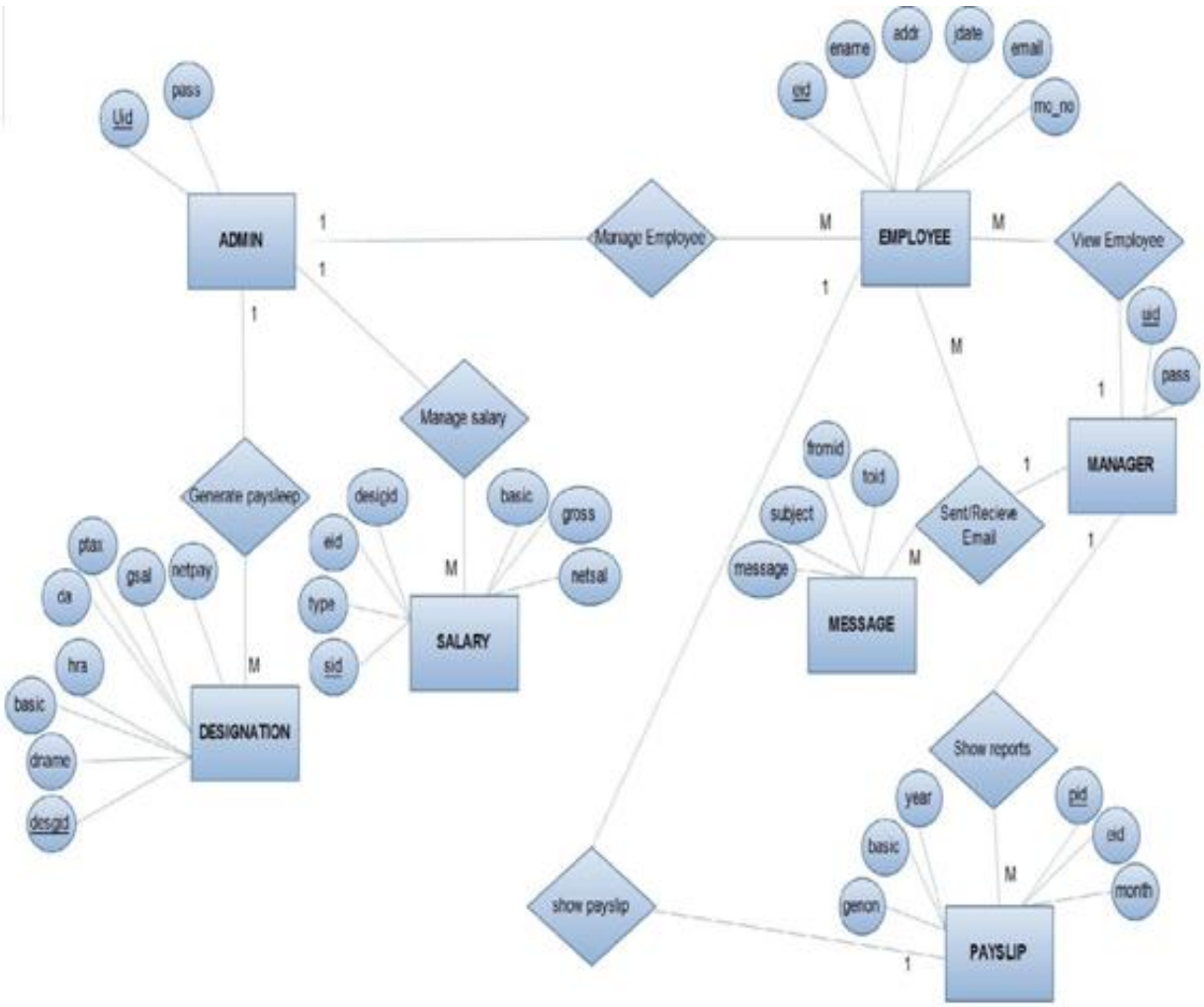
Salary Management System gives you the power to:

- Manage **Employee Information** Efficiently.
- Mark attendance of employees using Face Recognition.
- Define the **emoluments, deductions, leave** etc.
- Generate **Pay-Slip** at the convenience of a mouse click.
- Generate and Manage the **Payroll Processes** according to the **Salary Structure**
- assigned to the employee.
- Generate all the **Reports** related to employee, attendance/leave, payroll etc.
- Manage Cookies

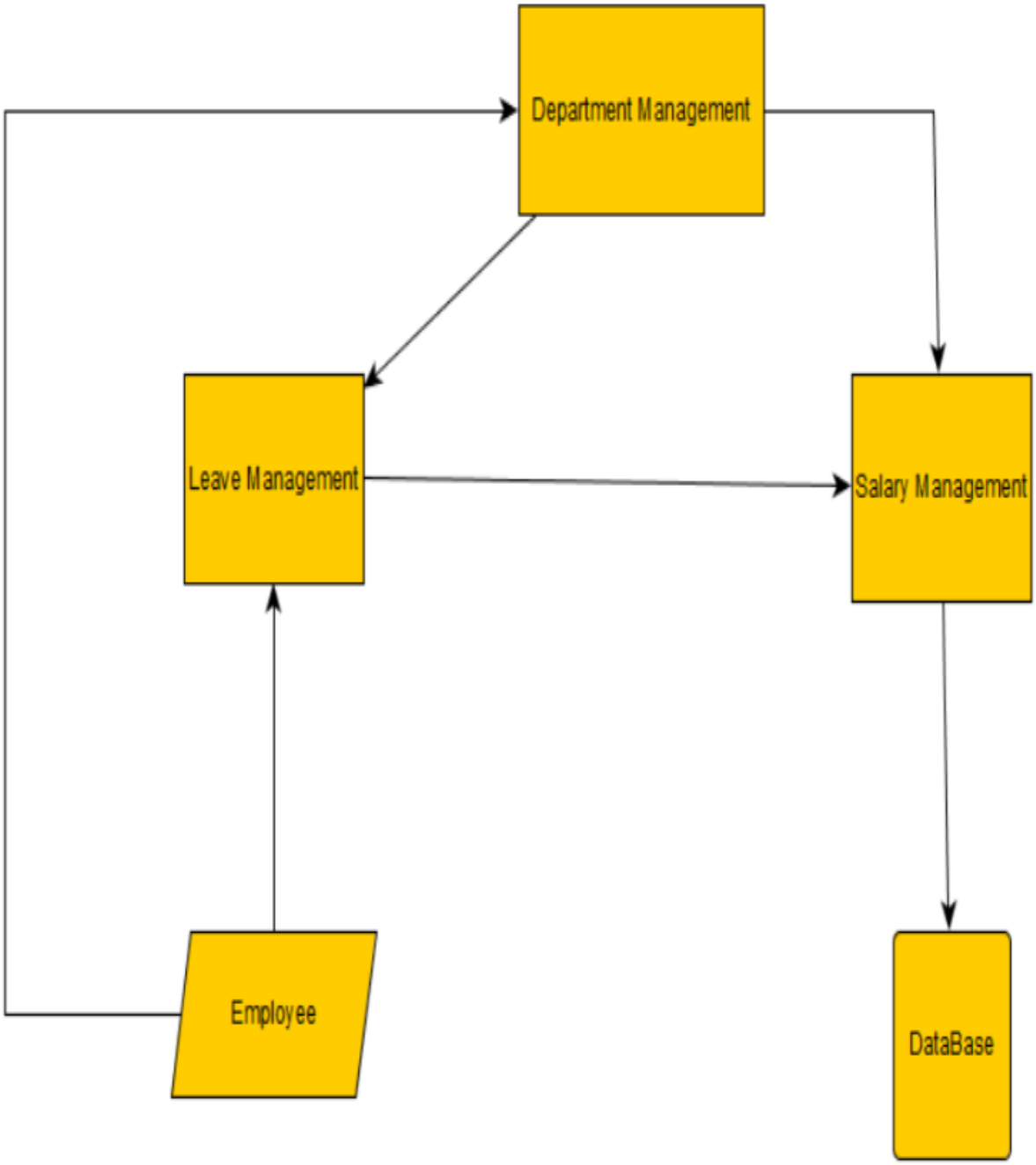


METHODOLOGY:

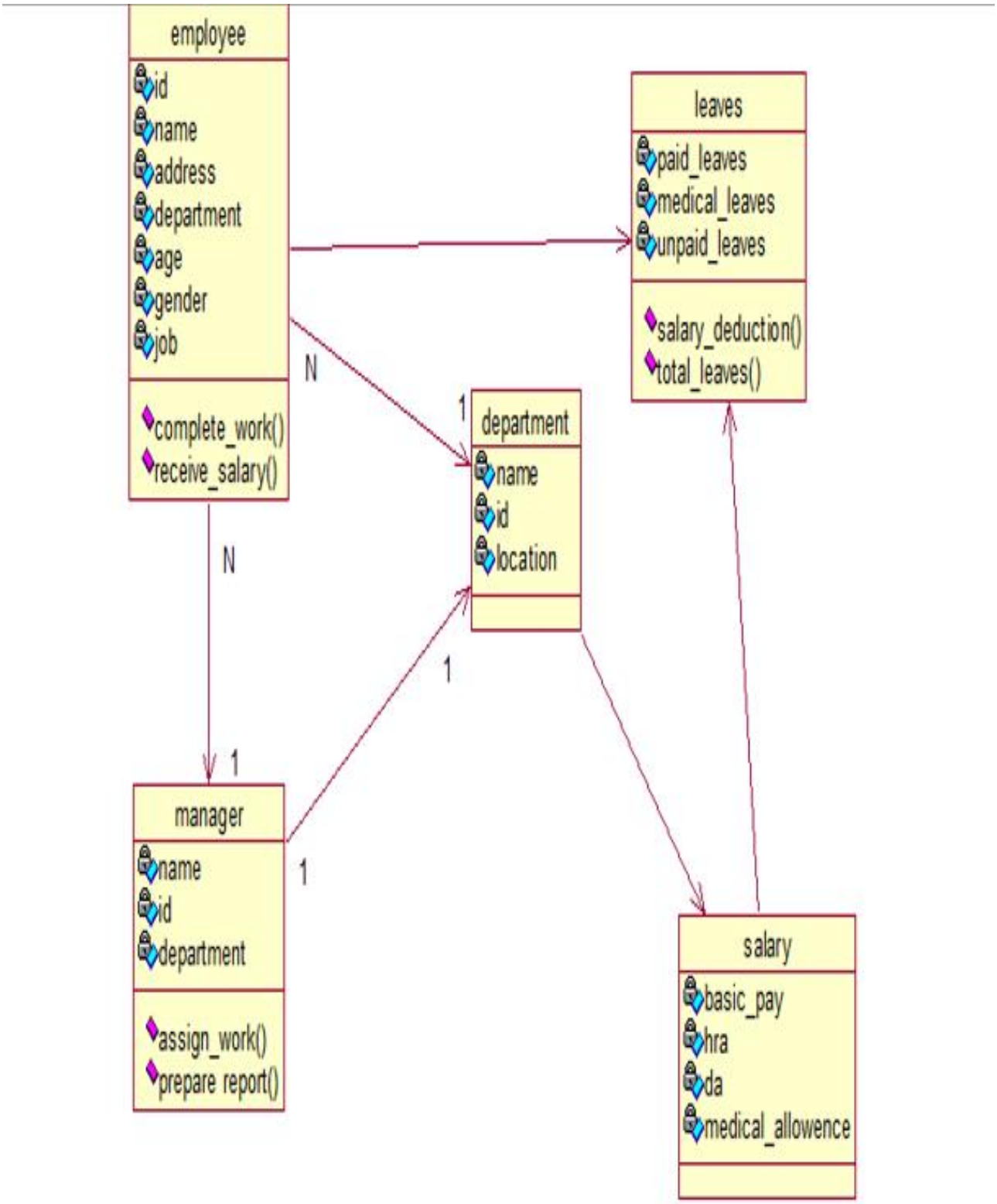
ER Diagram:



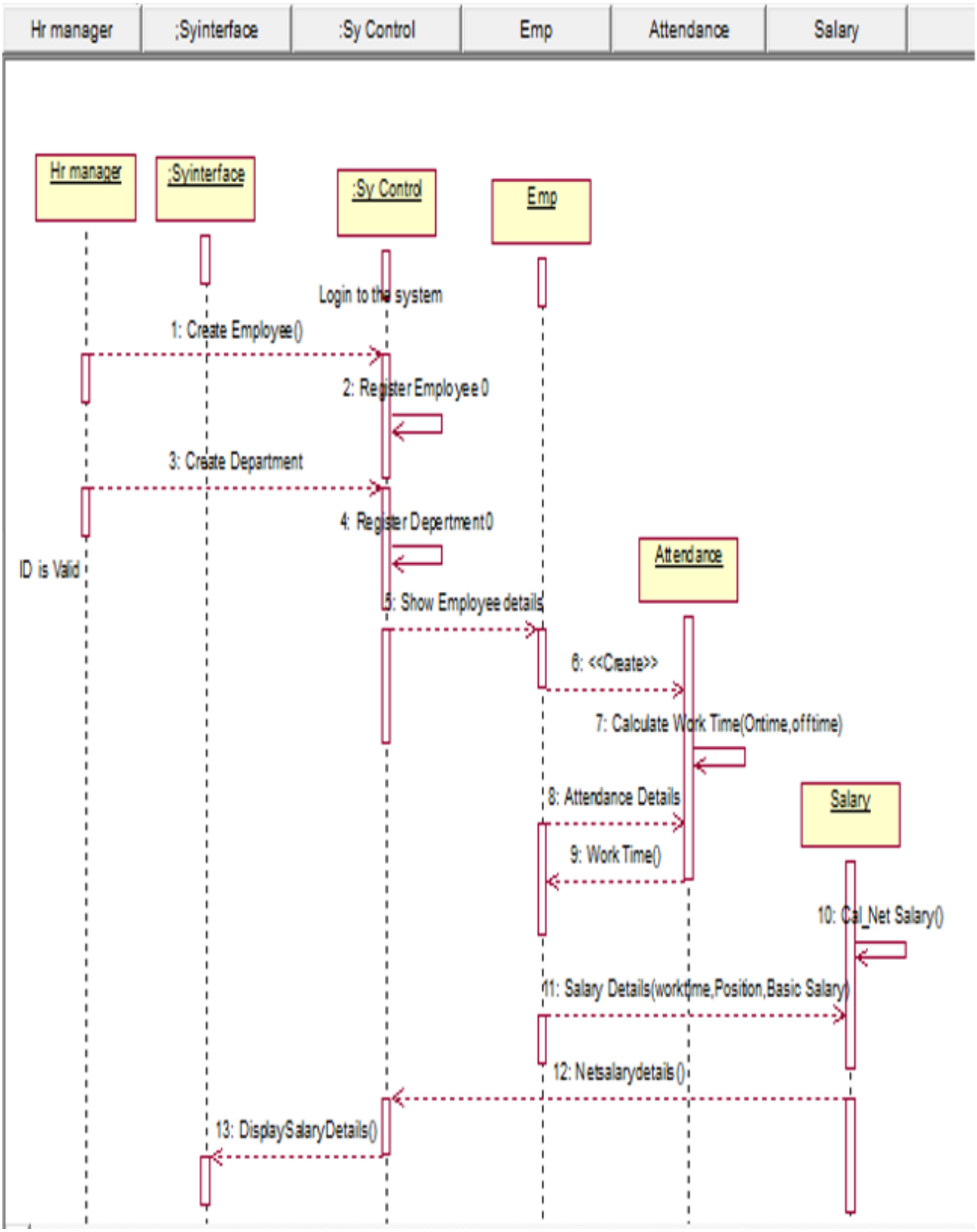
System Architecture: -



Class Diagram: -

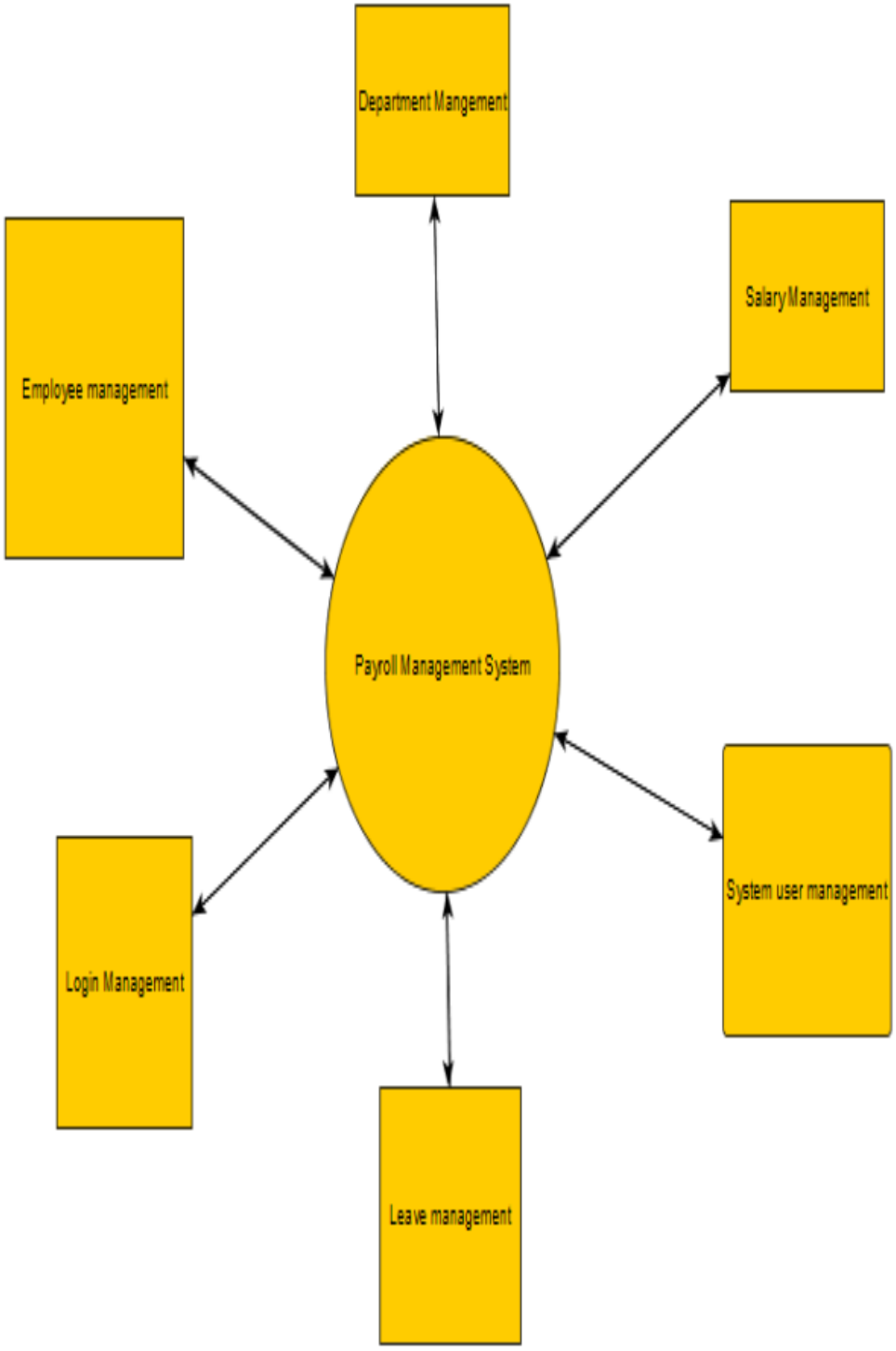


Sequence Diagram: -

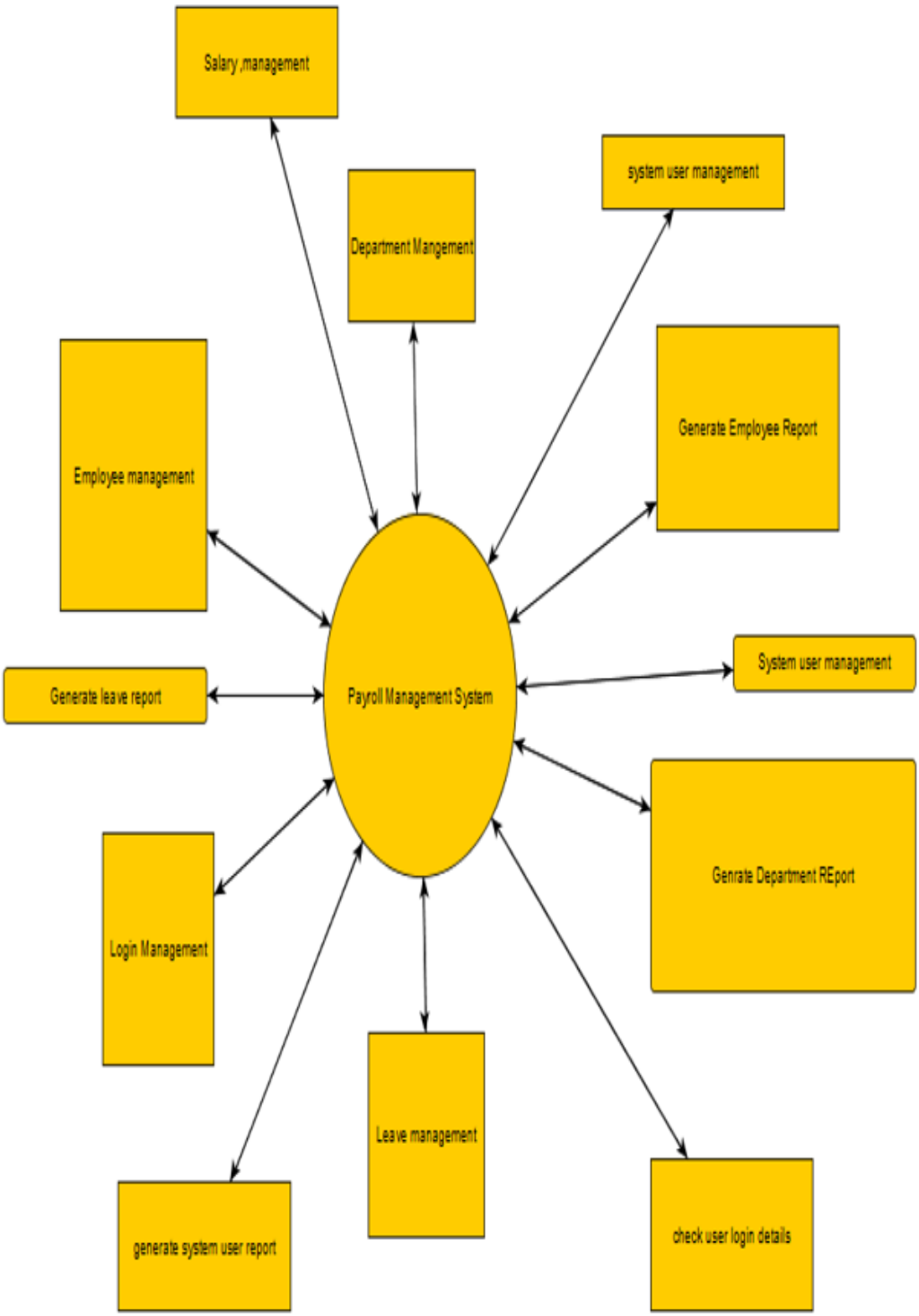


Data Flow Diagram:-

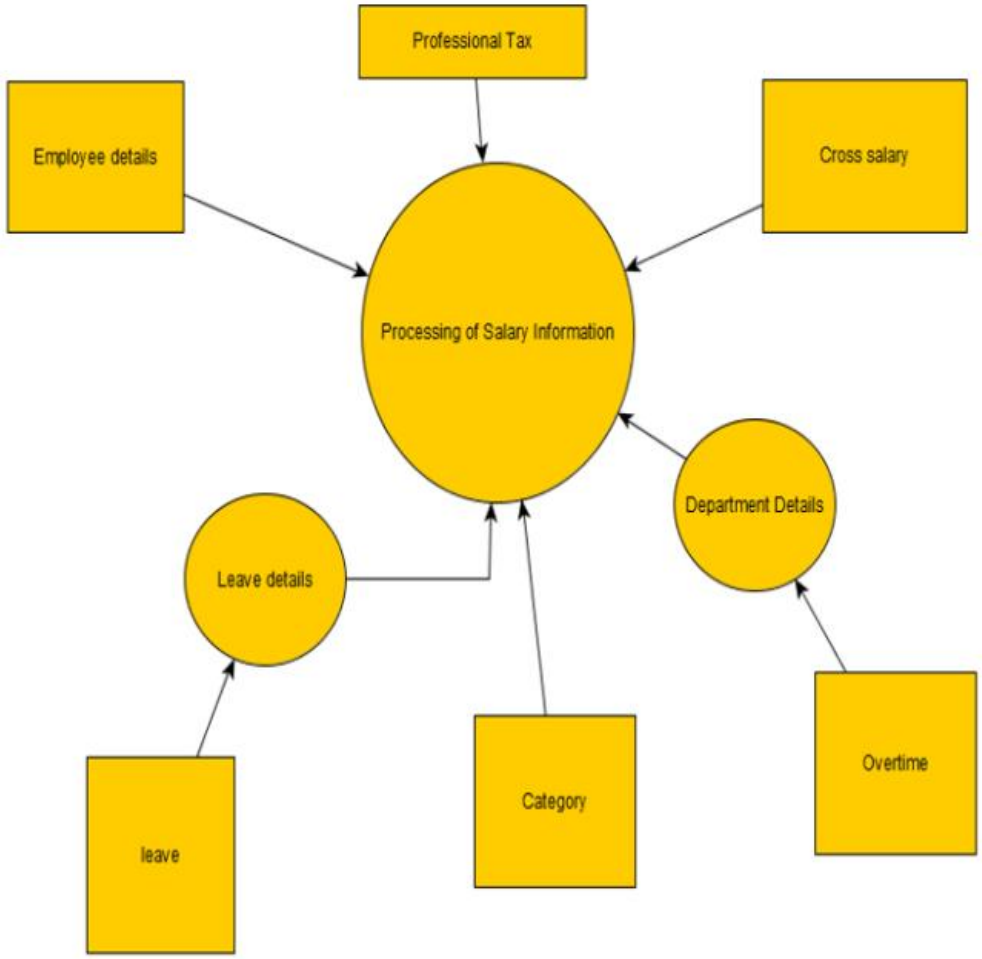
DFD(Level 0):-



DFD(Level 1):-



DFD(Level 2):-



Face recognition and detection:
For face recognition the steps involved are as stated below:
Collection of data ---- >Training process ----- >Testing process

The framework pre-forms the pictures. Eye coordinates are utilized to enlist the picture and the non-confront region is prohibited utilizing a curved cover. The grey histogram is then evened out finished the non-conceal territory. The calculation is prepared utilizing a subset of the pictures. A separation grid is gotten containing the separation between each match of pictures as a yield when the pre-prepared pictures are encouraged into the test calculation. By influencing the utilization of different settings for exhibition and test to picture sets and the acquired separation grid, the rank bends for the framework is computed. These can be figured for pre determined exhibition and test picture sets or by picking an arbitrary stages of one expansive set as test and display sets and computing the normal execution. The earlier technique is viewed as better in light of the fact that with that strategy it is less complex to quantify the calculation's execution under specific difficulties while the later is thought to be nearly more solid.

ALGORITHM FOR FACE RECOGNITION:

START

For detection:
gray ←cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
face_cascade ←cv2.CascadeClassifier('opencv-files/lbpcascade_frontalface.xml')
faces ← face_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5);
if (len(faces) is equal to 0):


```
return None, None
x, y, w, h = faces[0]
return gray[y:y+w, x:x+h], faces[0]
label ←dir_name.replace("s", ""))
subject_dir_path ← data_folder_path + "/" + dir_name
subject_images_names ← os.listdir(subject_dir_path)
for image_name in subject_images_names:
if image_name.startswith(".")
continue;
image_path ←subject_dir_path + "/" + image_name
image ← cv2.imread(image_path)
cv2.imshow(Training on image...", image)
face, rect ← detect_face(image)
if face is not None:
cv2.destroyAllWindows()
cv2.destroyAllWindows()
For train function:
dirs← os.listdir(data_folder_path)
faces ← []
labels ← []
 for dir_name in dirs:
if not dir_name.startswith("s"):
continue;
label ← int(dir_name.replace("s", ""))
subject_dir_path← data_folder_path + "/" + dir_name
subject_images_names ← os.listdir(subject_dir_path)
image_path ←subject_dir_path + "/" + image_name
image ←cv2.imread(image_path)
 cv2.imshow("Training on image...", image)
face, rect ←detect_face(image)
if face is not None:
faces.append(face)
labels.append(label)
cv2.destroyAllWindows()
cv2.destroyAllWindows()
 def prepare_training_data(data_folder_path):
label ← int(dir_name.replace("s", ""))
subject_dir_path ←data_folder_path + "/" + dir_name
subject_images_names ←os.listdir(subject_dir_path)
for image_name in subject_images_names:
if image_name.startswith("."):
continue;
image_path ←subject_dir_path + "/" + image_name
image ←cv2.imread(image_path)
cv2.imshow("Training on image...", image)
face, rect ← detect_face(image)
cv2.destroyAllWindows()
cv2.destroyAllWindows()
def prepare_training_data(data_folder_path):
dirs ←os.listdir(data_folder_path)
faces ← []
labels ← []
label ← dir_name.replace("s", "")
subject_dir_path ← data_folder_path + "/" + dir_name
```



```
subject_images_names ←os.listdir(subject_dir_path)
for image_name in subject_images_names:
if image_name.startswith("."):
continue;
image_path ← subject_dir_path + "/" + image_name
image ←cv2.imread(image_path)
cv2.imshow("Training on image...", image)
cv2.waitKey(100)
For recogniser function:
face, rect ←detect_face(image)
cv2.destroyAllWindows()
cv2.destroyAllWindows()
dirs ←os.listdir(data_folder_path)
faces ← []
labels ← []
label ←dir_name.replace("s", "")
subject_dir_path ←data_folder_path + "/" + dir_name
image_path ←subject_dir_path + "/" + image_name
image ←cv2.imread(image_path)
face, rect ← detect_face(image)
cv2.destroyAllWindows()
cv2.destroyAllWindows()

END
```

Sample C++ Code:

```
#include <iostream.h>

#include <fstream.h>

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include <ctype.h>

#include <conio.h>

#include <dos.h> class MENU

{

public :

void MAIN_MENU(void) ;

void EDIT_MENU(void) ; void INTRODUCTION(void) ;

}M ;

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  RECORDNO(int) ; int FOUND_CODE(int) ;
void DISPLAY_RECORD(int) ; int VALID_DATE(int, int, int) ; int code, dd, mm, yy ;
char name[31], address[41], phone[10], desig[21] ; char grade, house, conveyence ;
float loan, basic ;
} E;
void MENU :: INTRODUCTION ()
{ window(0,0,640,320);
gotoxy(30,10);cout<<       "WELCOME";
gotoxy(25,11);cout<<       "****************** ";
gotoxy(25,12);cout<<       " ";
gotoxy(8,13);cout<< "      THIS IS PAYROLL MANAGEMENT SYSTEM FOR SOFTWARE ENGINEERING ";
 gotoxy(8,14);cout<<" ";
gotoxy(8,15);cout<<" ";
gotoxy(8,16);cout<<" ";
gotoxy(8,17);cout<<" "; gotoxy(40,18); gotoxy(2,25);
cout<<"PRESS ANY KEY TO CONTINUE....";
getch();
clrscr();
gotoxy(8,2);
cout<<"            TEAM MEMBERS \n\n\n\n\n"; cout<<"      AKARSH GUPTA                              16BCE2153
\n";
gotoxy(2,25);
cout<<"PRESS ANY KEY TO CONTINUE ....";
getch();
clrscr();
}
void MENU :: MAIN_MENU(void)
```



```
{
char ch ; while (1)
{
 clrscr() ;
gotoxy(28,8) ;
cout <<"PAYROLL-MANAGEMENT\n " ;
gotoxy(20,9) ; cout<<"**************************************"; 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') E.NEW_EMPLOYEE() ;
else
 if (ch == '2')
E.DISPLAY() ;
else
if (ch == '3')
E.LIST() ;
else
if (ch == '4') E.SALARY_SLIP() ;
else
if (ch == '5') EDIT_MENU() ;
}
}
```

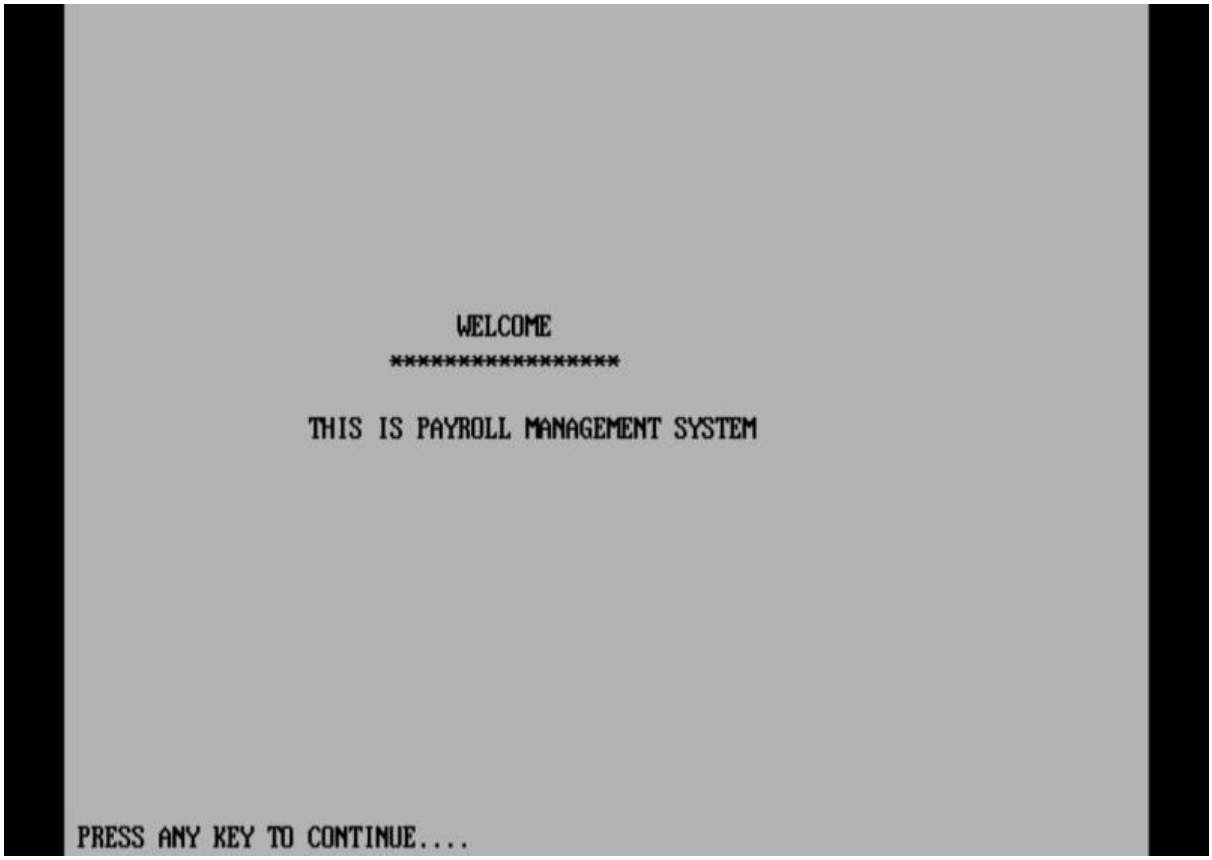


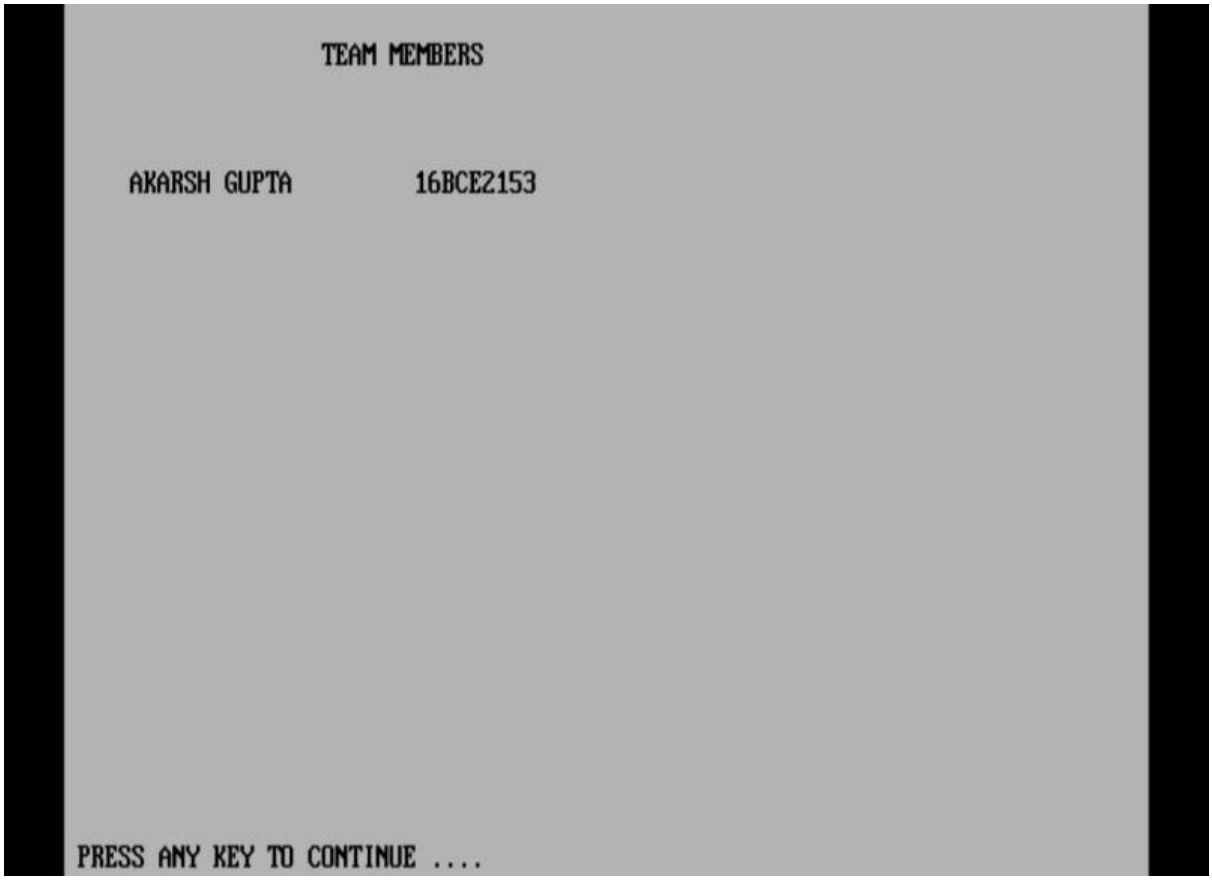
```
void MENU :: EDIT_MENU(void)
{
char ch ; while (1)
{
clrscr() ;
gotoxy(31,9) ;
cout <<"E D I T M E N U" ;
gotoxy(30,13) ;
cout <<"1: DELETE RECORD" ;
gotoxy(30,14) ;
cout <<"2: MODIFY RECORD" ;
gotoxy(30,15) ; cout <<"0: EXIT" ;
gotoxy(30,17) ;
cout <<"ENTER YOUR CHOICE :" ;
ch = getch() ;
if (ch == 27 || ch == '0') break ;
else
if (ch == '1')
E.DELETION() ;
else
if (ch == '2') E.MODIFICATION() ;
}
}
void EMPLOYEE :: ADD_RECORD(int ecode, char ename[31], char eaddress[41], char ephone[10], int d, int m, int y,
char edesig[21], char egrade, char ehouse, 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 = ehouse ; conveyence = econv ; loan = eloan ;
basic = ebasic ;
file.write((char *) this, sizeof(EMPLOYEE)) ; file.close() ;
}
void EMPLOYEE :: MODIFY_RECORD(int ecode, char ename[31],char eaddress[41],
```

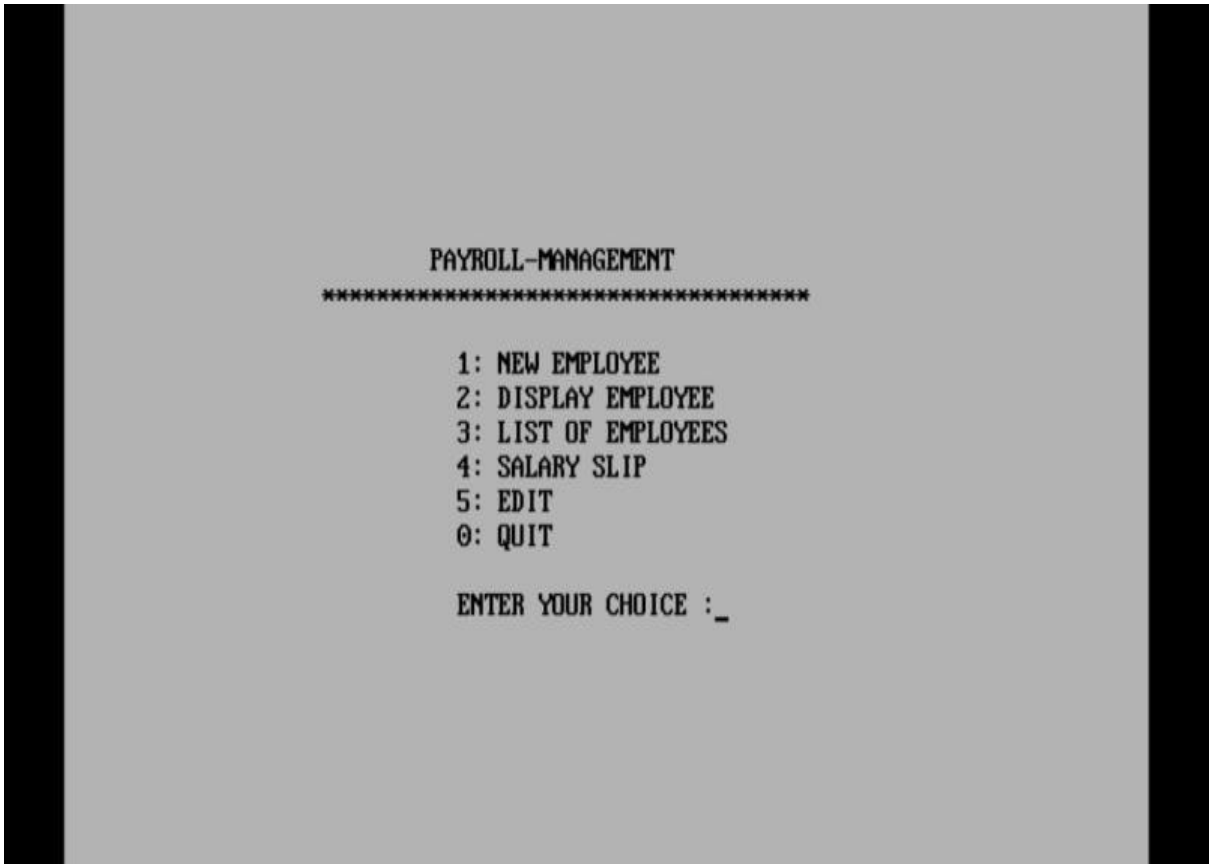


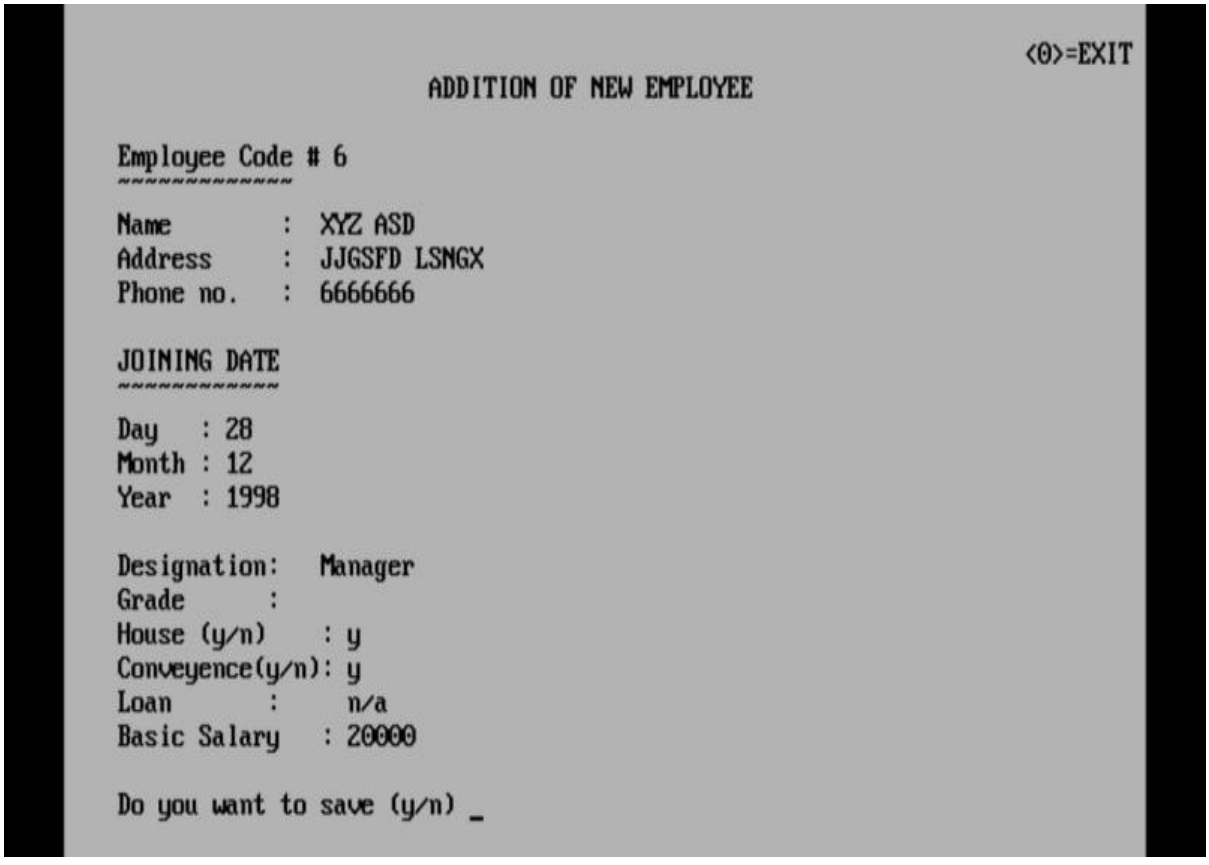
```
char ephone[10], char edesig[21], char egrade, char ehouse, char econv, float eloan, float ebasic)
{
int recno ;
recno = RECORDNO(ecode) ; fstream file ;
file.open("EMPLOYEE.DAT", ios::out | ios::ate) ;
strcpy(name,ename) ; strcpy(address,eaddress) ; strcpy(phone,ephone) ; strcpy(desig,edesig) ; grade = egrade ;
house = ehouse ; conveyence = econv ; loan = eloan ;
basic = ebasic ; int location ;
location = (recno-1) * sizeof(EMPLOYEE) ; file.seekp(location) ;
file.write((char *) this, sizeof(EMPLOYEE)) ; file.close() ;
}
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() ;}
```

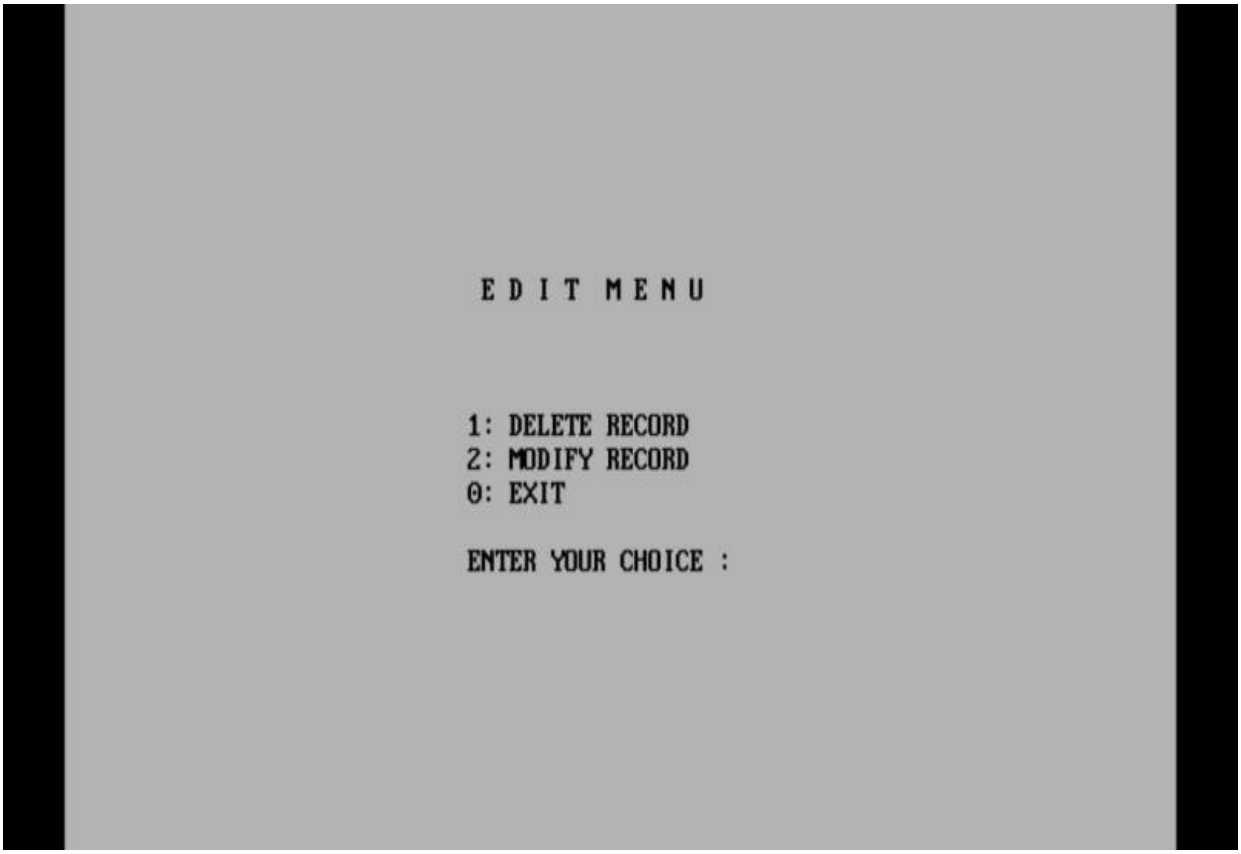

Snapshots of Program:

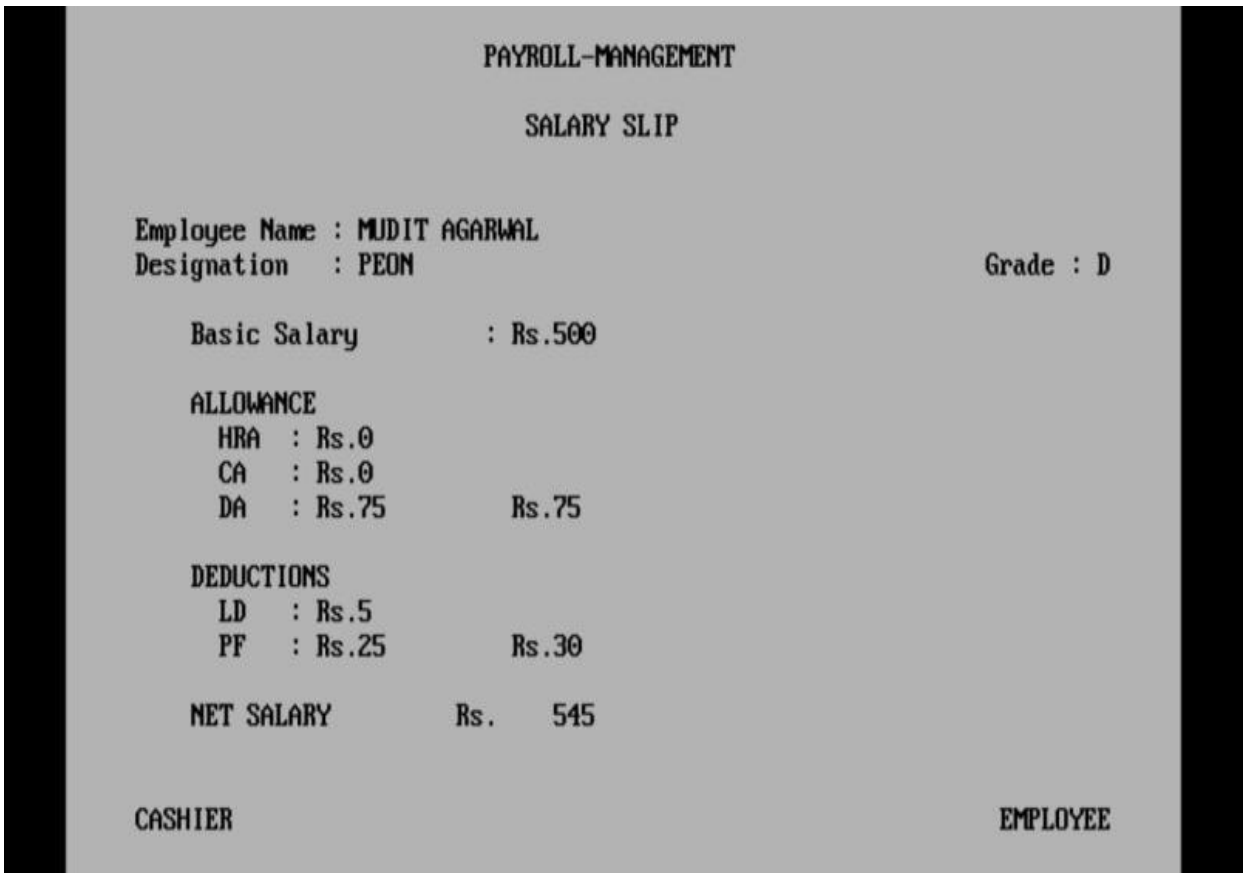


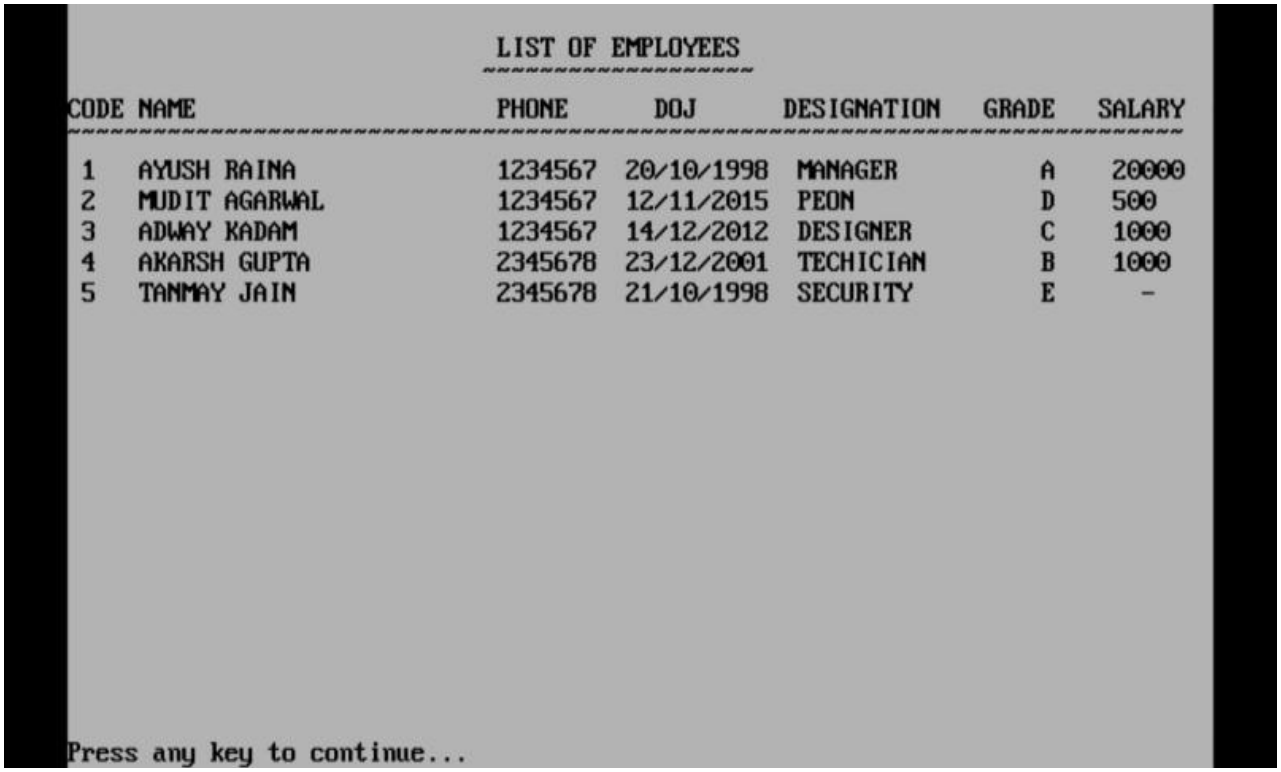








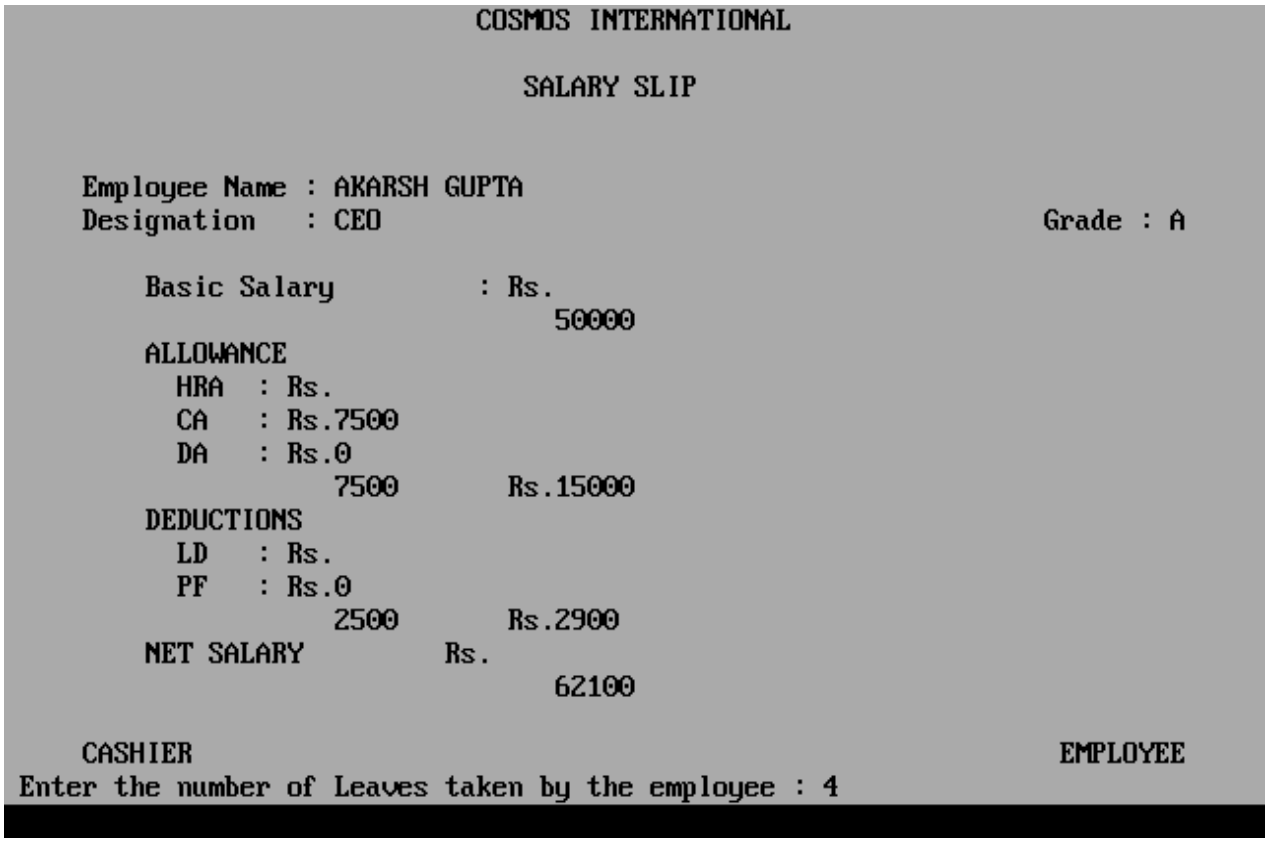




LIST OF EMPLOYEES

CODE	NAME	PHONE	DOJ	DESIGNATION	GRADE	SALARY
1	AYUSH RAINA	1234567	20/10/1998	MANAGER	A	20000
2	MUDIT AGARWAL	1234567	12/11/2015	PEON	D	500
3	ADWAY KADAM	1234567	14/12/2012	DESIGNER	C	1000
4	AKARSH GUPTA	2345678	23/12/2001	TECHICIAN	B	1000
5	TANMAY JAIN	2345678	21/10/1998	SECURITY	E	-

Press any key to continue...



COSMOS INTERNATIONAL

SALARY SLIP

Employee Name : AKARSH GUPTA
Designation : CEO Grade : A

Basic Salary : Rs.
50000

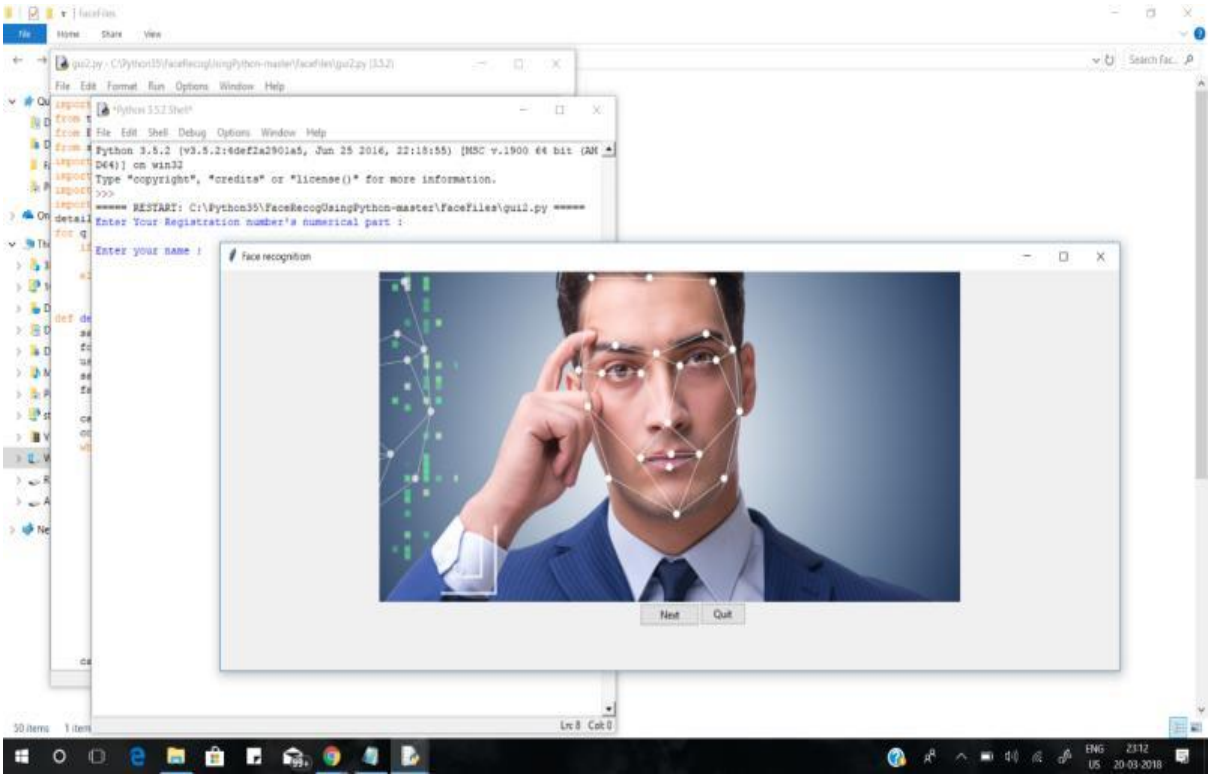
ALLOWANCE
HRA : Rs.
CA : Rs.7500
DA : Rs.0
7500 Rs.15000

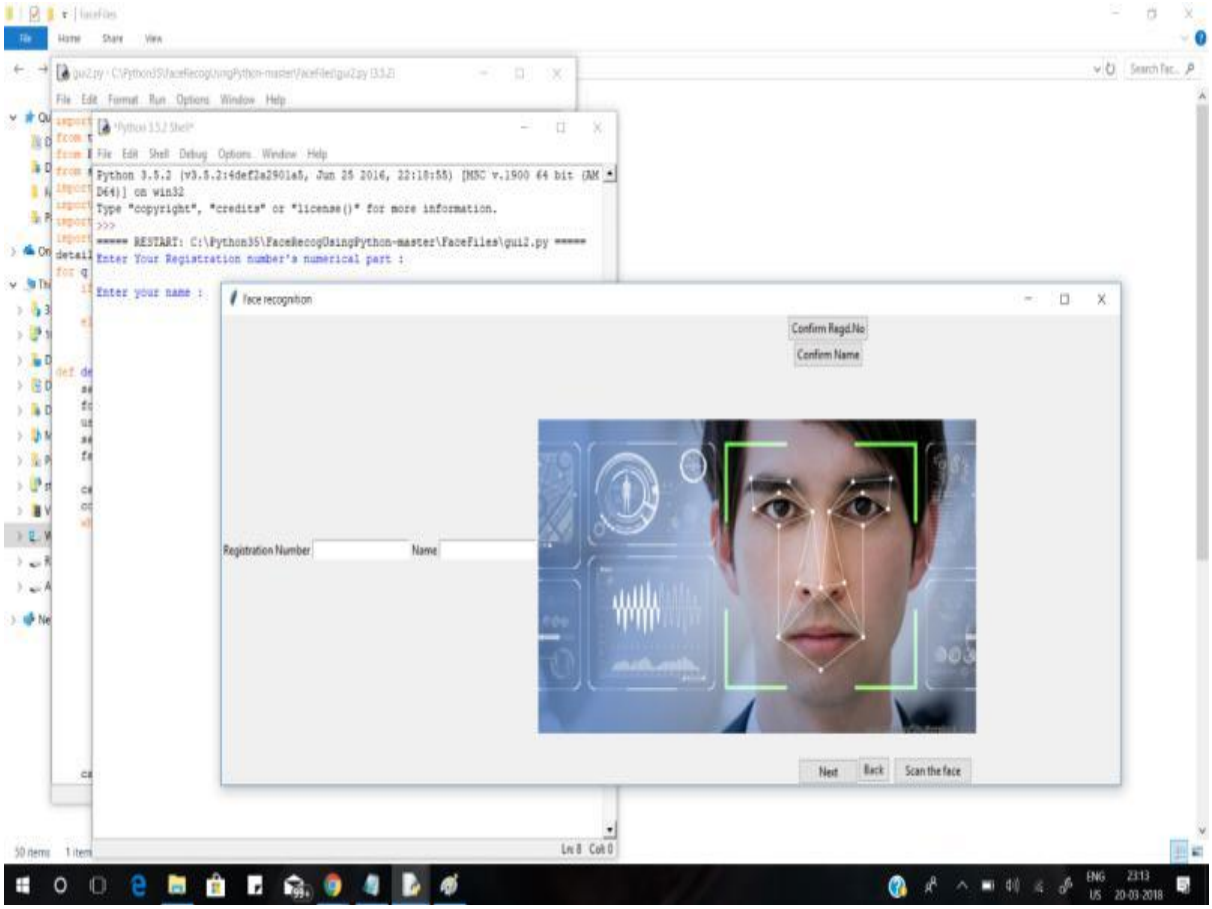
DEDUCTIONS
LD : Rs.
PF : Rs.0
2500 Rs.2900

NET SALARY Rs.
62100

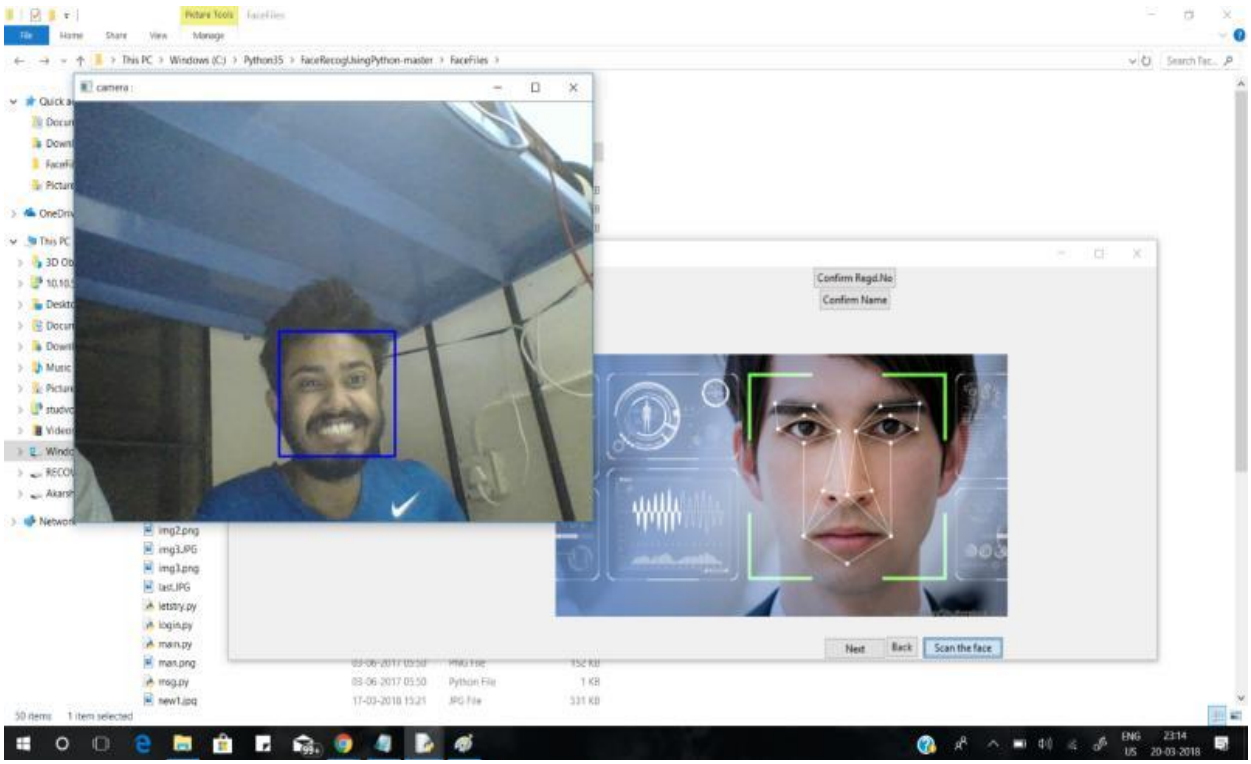
CASHIER EMPLOYEE

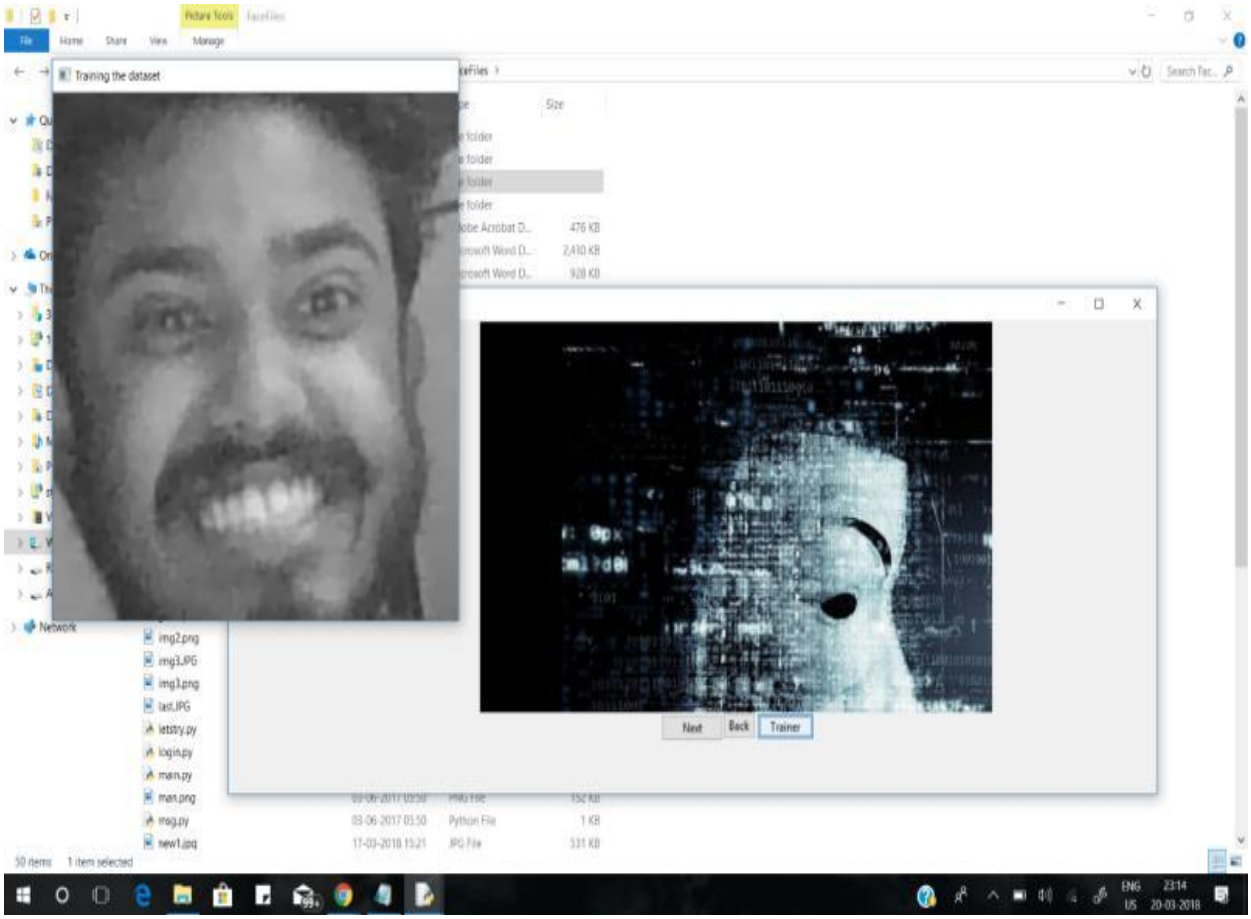
Enter the number of Leaves taken by the employee : 4

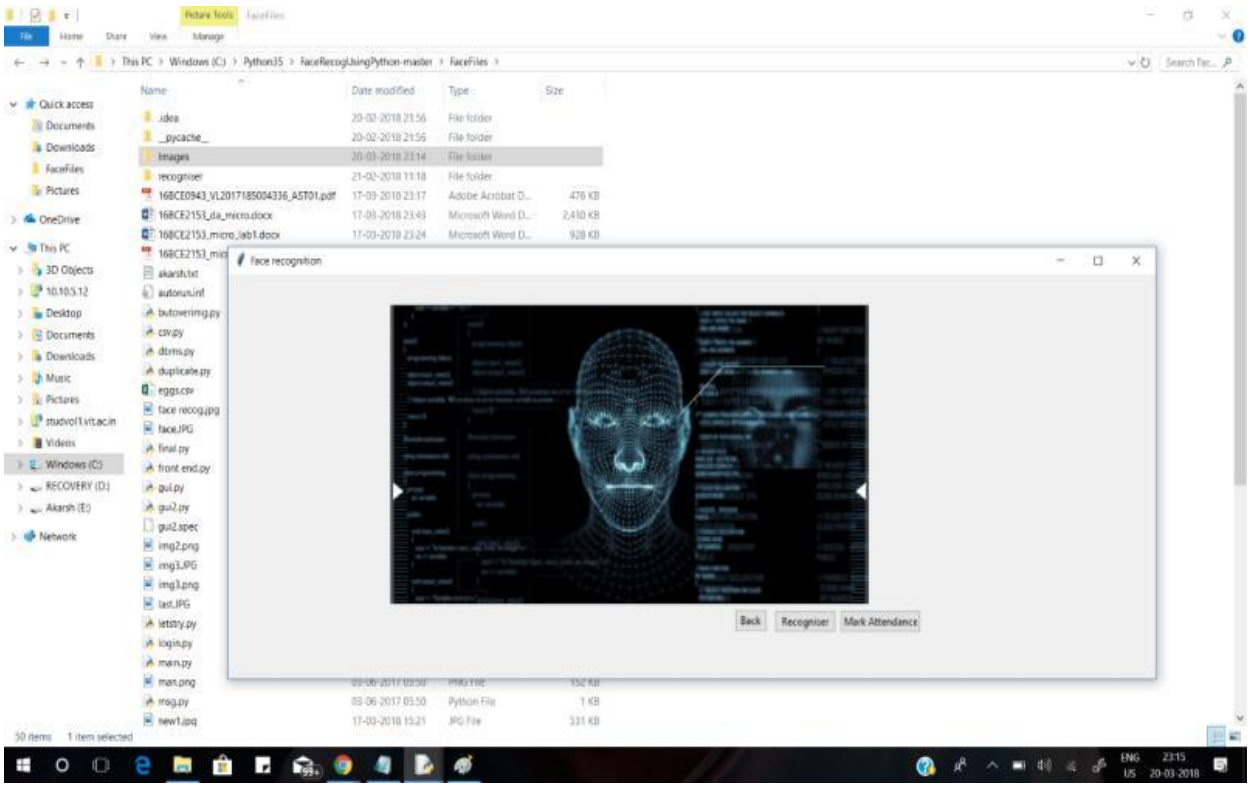


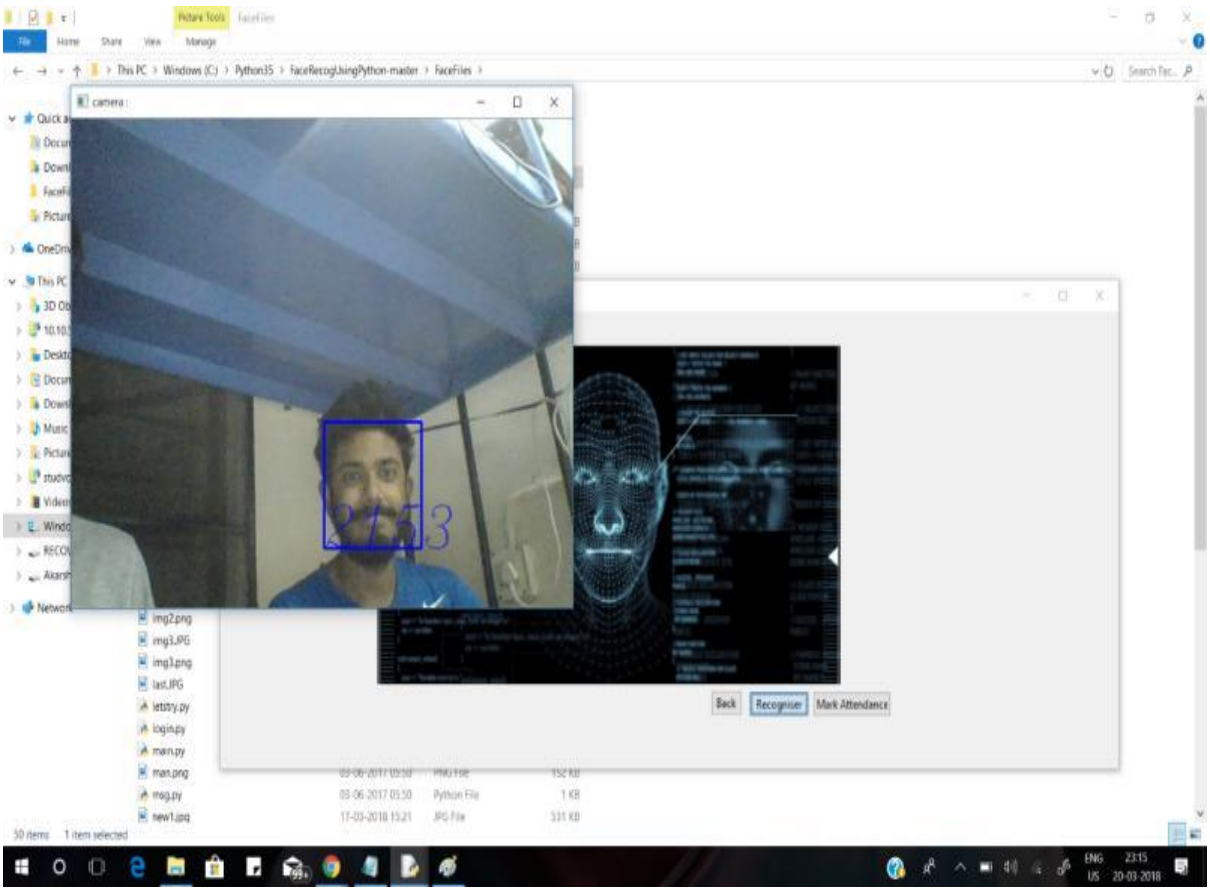


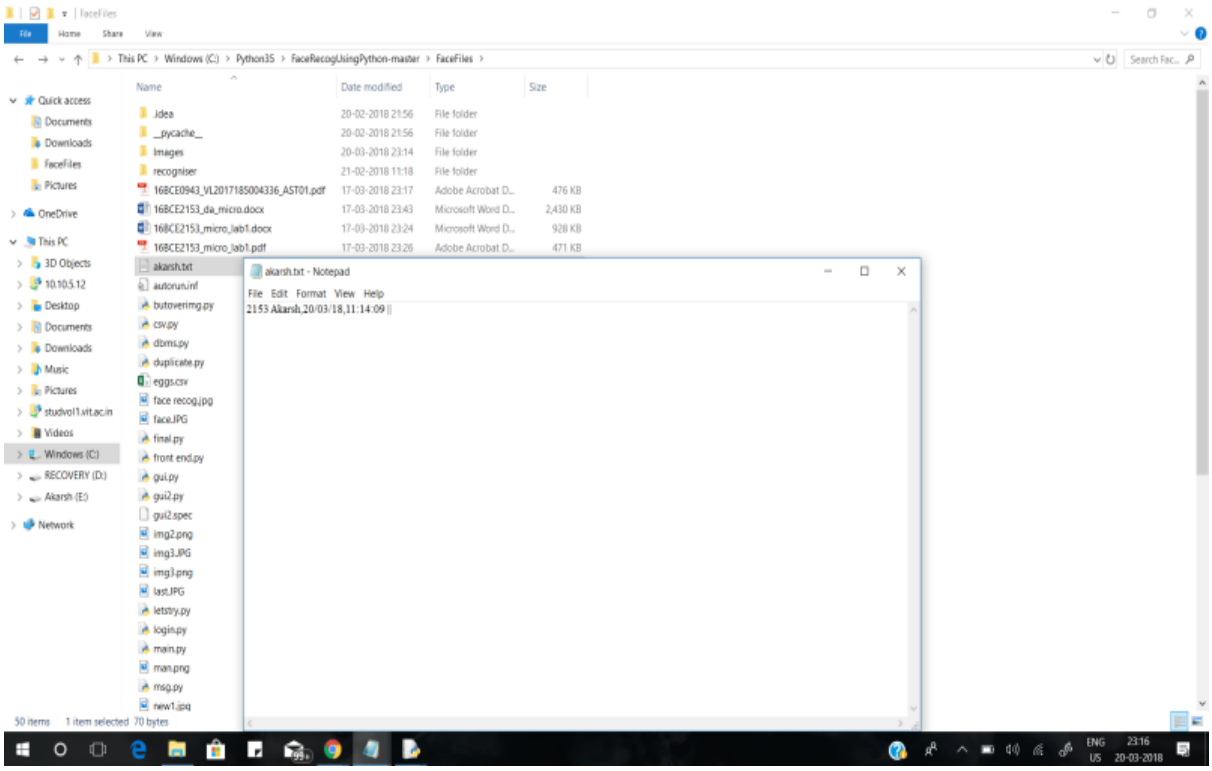
Face Recognition Snapshots:











ACKNOWLEDGEMENT

We would like to thank VIT University for providing us the opportunity to carry out this project. We also thank our batch mates and seniors in helping us to carry out our project work.
We thank Prof. Prabu S to be our project guide and guided us at various stages in completing the project. Without his support it would be very difficult for us to do the project. We would like to pay our gratitude to Sir for sharing his pearls of wisdom with us during the project.

Conclusion:

We have successfully implemented a company payroll management with a basic GUI implemented in C++ that generates salary of employee of a particular department with the ability to generate all the details of the desired employee. We have made use of concepts of database like Normalization, Primary Key and Foreign key concepts. Our Database is implemented on C++.We have used pointers to connect to the backend and successfully implemented it.

We have also used face recognition to mark attendance of employees and store it in a file with exact date and time at which the face of the employee was detected along with his/her name and employee id.

