C H A P T E R - 1
INTRODUCTION

1. MOTIVATION

Nowadays our college account staff is taking more time to generate the salary slip of an employee and they were doing it manually and submitting them in the form of hard copies. So, our application makes the task simple by sending the salary slip in the form of soft copies through E-mail, based on their monthly attendance and holidays. This will reduce workflow, time and human efforts.

2. PROBLEM DEFINITION

To prevent data redundancy and misuse of data we have created an application where admin can generate salary slip of an employee and send it to them through e-mail automatically.

The aim of this proposed system is to contribute to the goal of achieving the database management system that keeps the records of employees in the respective departments, thereby aligning the motivation of this project.

OBJECTIVE OF THE PROJECT

- ➢ Main objective is to generate the salary slip of an employee and sending in the form of soft copies through E-mail automatically.
- ➢ This salary slip maintains all the information about payments, employee, salary details.
- ➢ The project is totally built at administrative end and thus only the administrator is guaranteed the access.
- ➢ The purpose of this project is to build an application program to reduce the manual work for managing the payments, Salary employee, Working Points.

3. LIMITATIONS OF PROJECT

- ➢ As this is a desktop application, only administrator have access for this application.
- ➢ This application needs constantly backup the salary data.
- ➢ Access to this application is Limited.

4. ORGANISATION OF DOCUMENTATION

4.1. Feasibility Study

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

¼ Technical Feasibility

¼ Operation Feasibility

¼ Economic Feasibility

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

¼ Does the necessary technology exist to do what is suggested?

¼ Do the proposed equipment 's have the technical capacity to hold the data required to use the new system?

¼ Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

¼ Can the system be upgraded if developed?

¼ Are there technical guarantees of accuracy, reliability, ease of access and data security?

Operation Feasibility

The operational feasibility includes User friendly, reliability, security, portability, availability, and maintainability of the software used in the project.

Economic Feasibility

Analysis of a project costs and revenue to determine whether it is logical and possible to complete.

C H A P T E R - 2
LITERATURE SURVEY

1. INTRODUCTION

This system is to generate an automated salary slip of an employee which is based on their monthly attendance that is fetched from excel sheet through biometric machine and monthly salary. This salary will be calculated which contains attributes like hra, da, lop etc,.

2. EXISTING SYSTEM

In the existing system, the admin has to do the process manually and has to generate the salary slip in the form of hardcopies. This will take more time and work effort to complete.

3. DISADVANTAGES OF EXISTING SYSTEM

➢ Increases human effort and workflow.
➢ More data redundancy and large usage of data.

4. PROPOSED SYSTEM

In the Proposed System we have implemented the system efficiently that admin can easily generates salary slip of every employee and sends it to them through email which will be done automatically.

5. ADVANTAGES OF EXISTING SYSTEM

¼ Web applications are present.

CHAPTER-3
ANALYSIS

1. INTRODUCTION

Our project title, "EMPLOYEE SALARY GENERATION SYSTEM" is meant for reducing the time, human effort and workflow by generating the salary slip of an employee and send it to them through E-mail automatically based on their monthly attendance, leaves and holidays.

2. HARDWARE AND SOFTWARE DESCRIPTION

Hardware Requirements:

- ➢ At least 1 GB Hard Disk
- ➢ 256 MB RAM
- ➢ Intel Dual Core Processor

Software Requirements:

- ➢ Windows 10,8th generation
- ➢ JUPYTER,PYTHON IDLE
- ➢ MYSQL

3. CONTENT DIAGRAM OF PROJECT

WORKING OF THE PROJECT:

The Working in this project is simple and easily understandable. The admin is provided with default user id and password. He can login to the portal and can perform his operations. Admin will be able to add/update/delete details of the employee.

The admin fetches the attendance in the system that has generated in excel sheet from biometric machine and updates it in the database. By analyzing all the leaves and holidays that the employee has taken in the month our application calculates the salary information and deduct the money from the salary and generates the salary slip. When the admin clicks “view results” it will display salary slip of every employee.

CHAPTER-4
DESIGN

1. INTRODUCTION

To represent the working procedure of the system, we have used Unified Modeling Language (UML) diagrams. UML diagrams are implemented using Rational Rose software.

2. ER/UML DIAGRAMS

ER DIAGRAM:

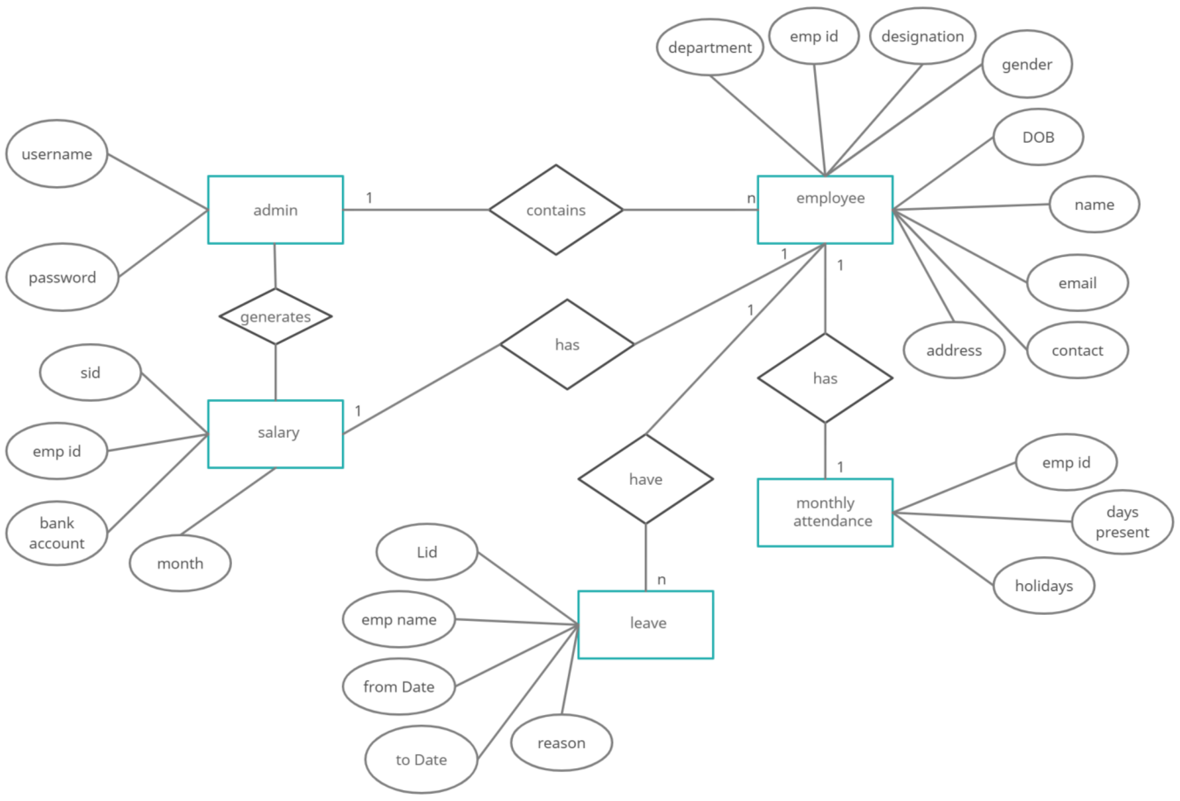


Figure 4.2.1: ER diagram

Usecase Diagram:

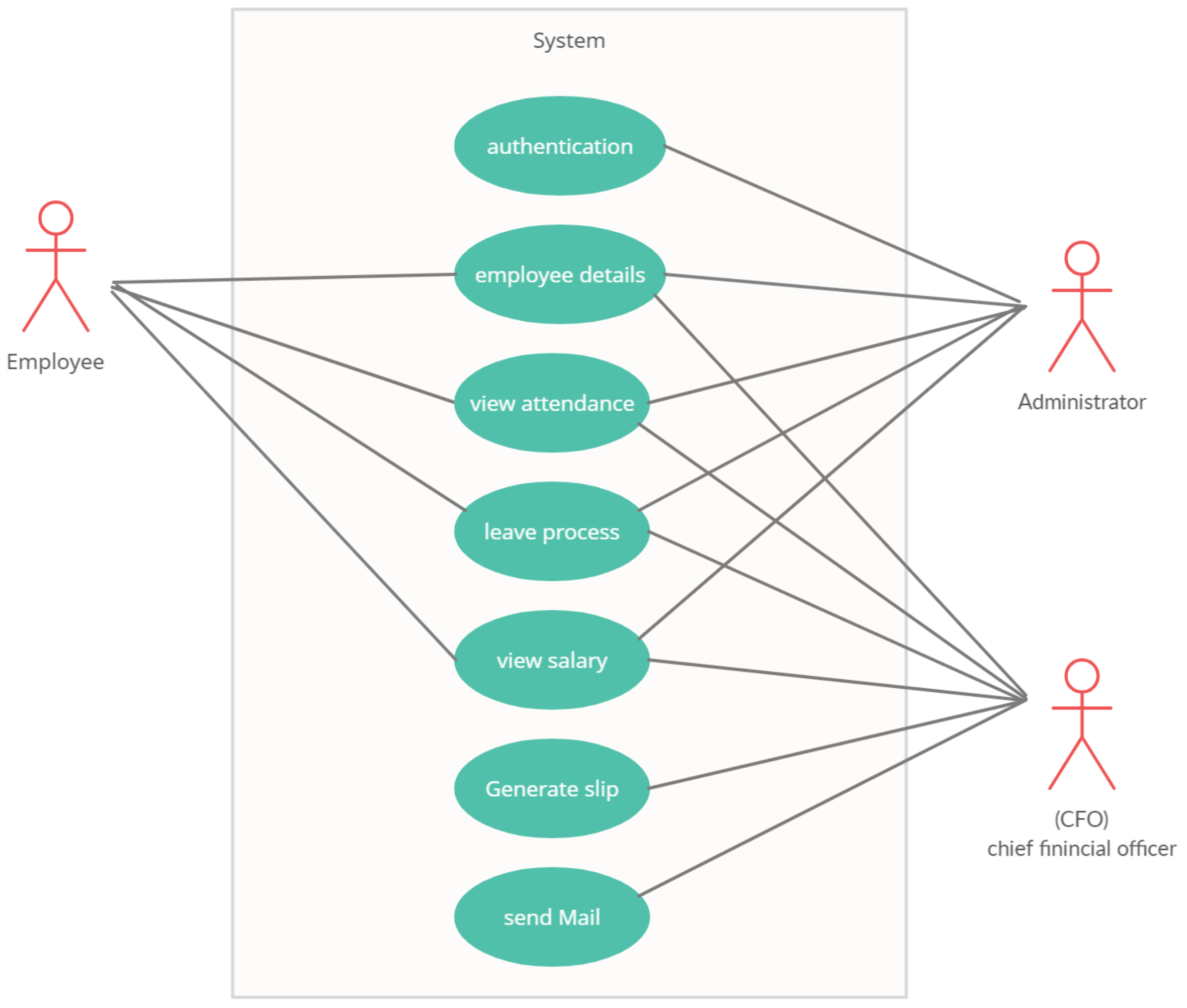


Figure 4.2.2: Use case diagram

Class Diagram:

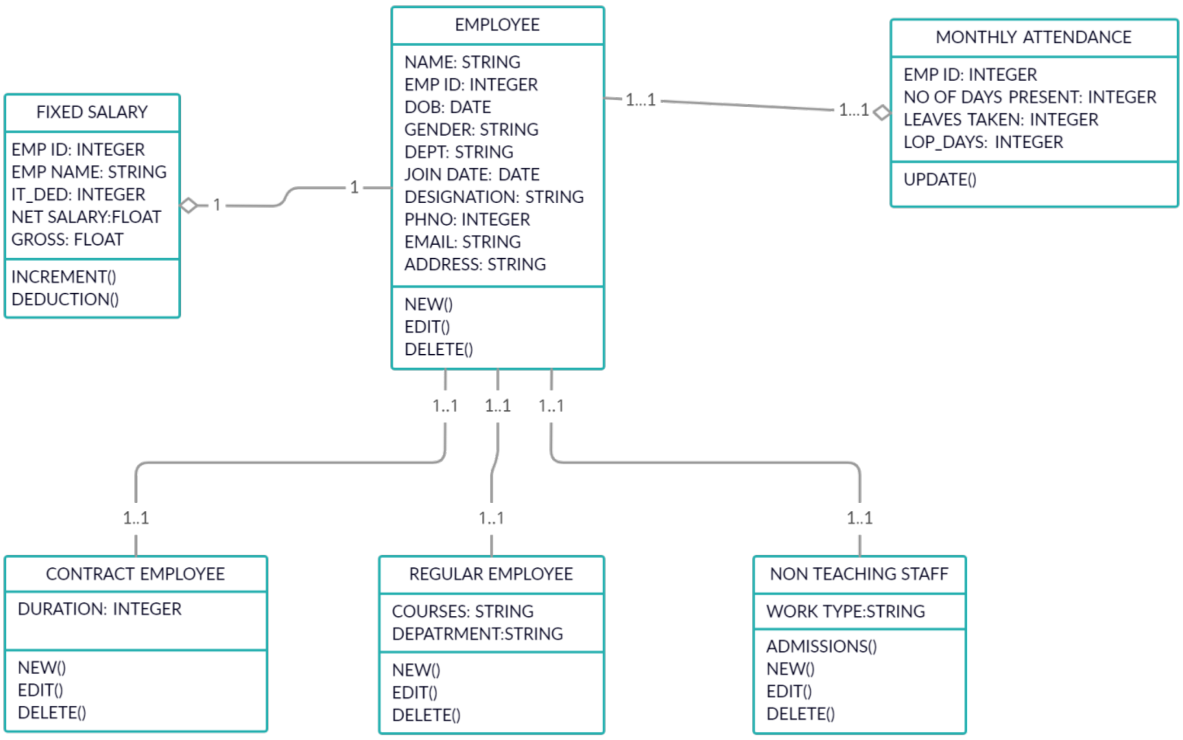


Figure 4.2.3: Class diagram

ActivityDiagram:

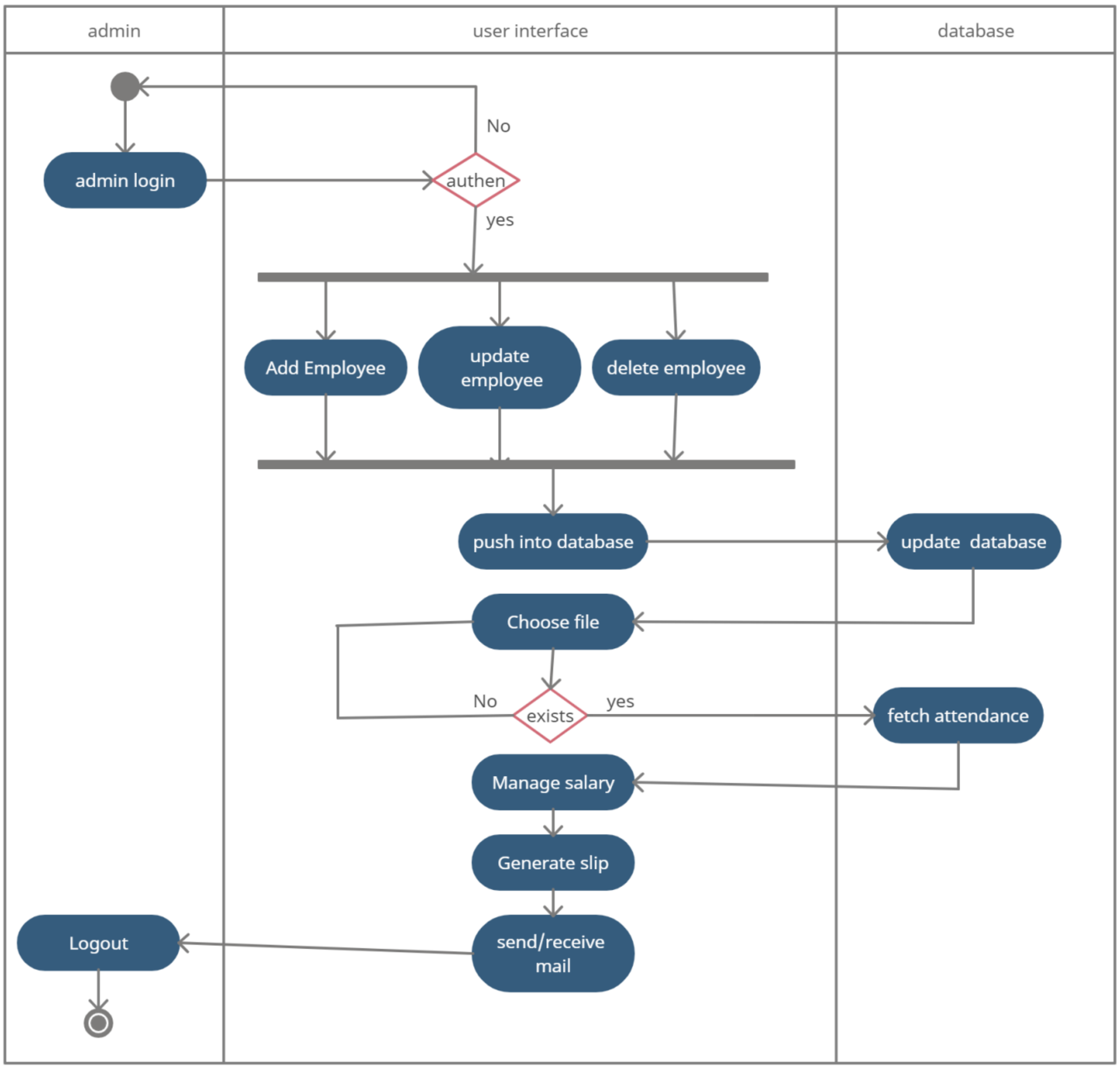


Figure 4.2.4: Activity diagram

Sequence Diagram:

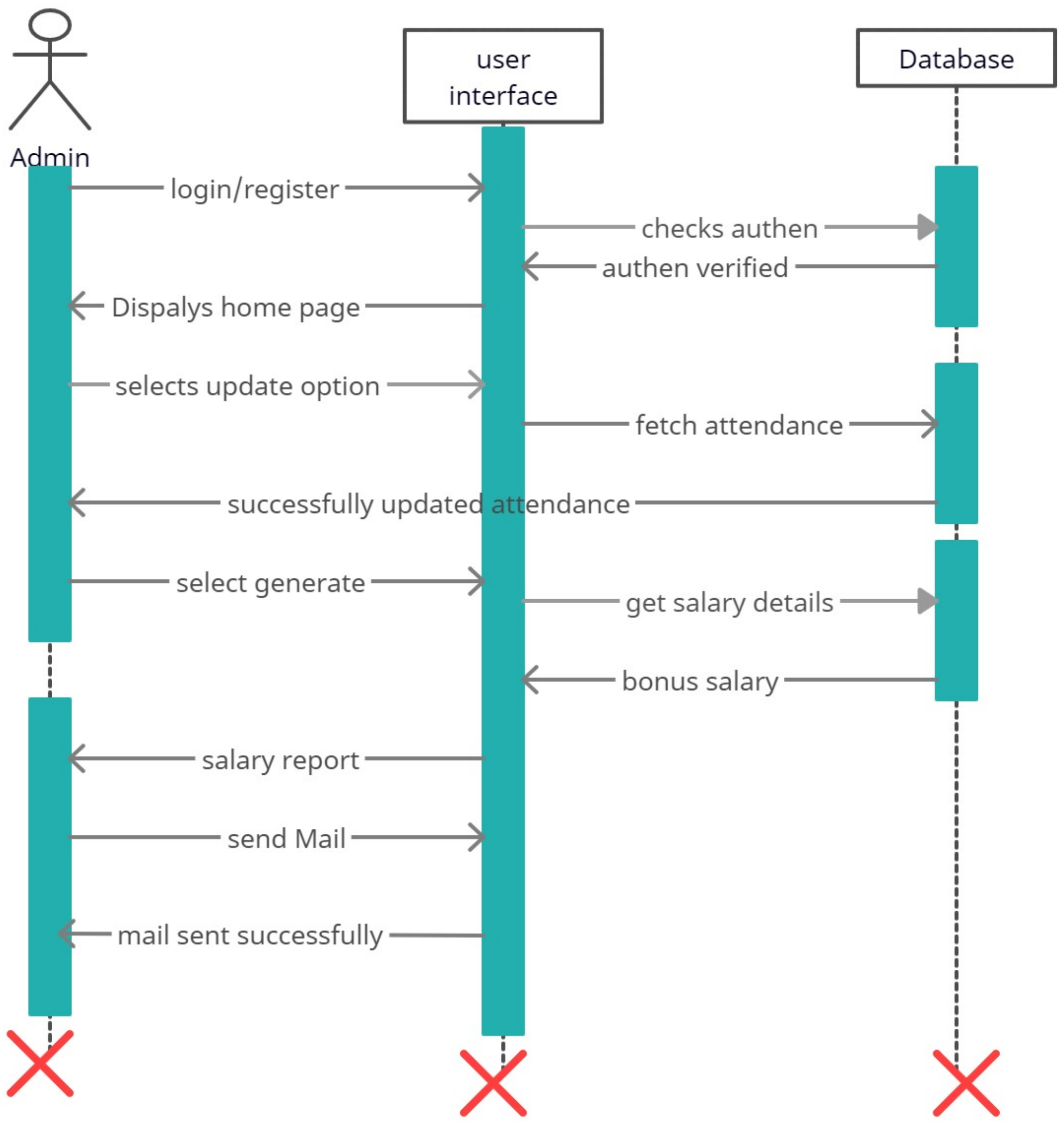


Figure 4.2.5: Sequence diagram

State chart Diagram:

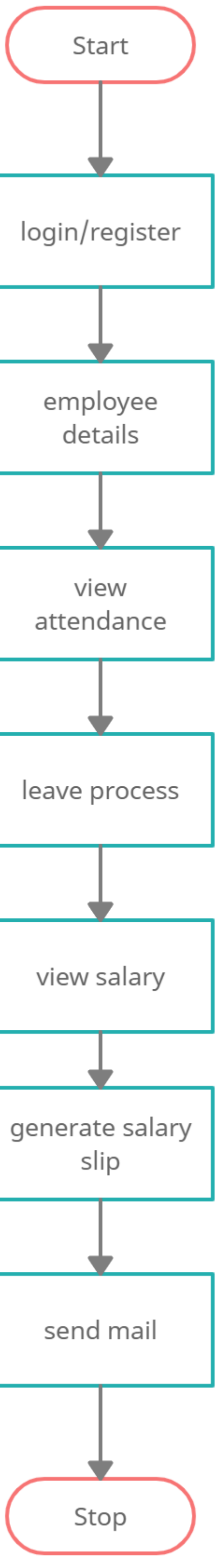


Figure 4.2.6: State Chart diagram

Collabaration Diagram:

`

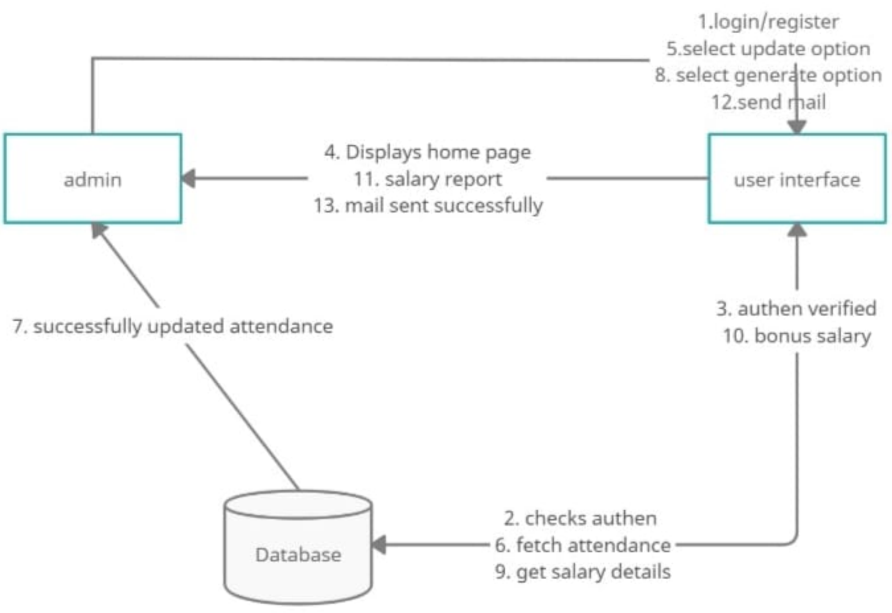


Figure 4.2.7: Collaboration diagram

3. MODULE DESIGN AND ORGANIZATION

Creating credentials

If the administrator wants to view the details of any employee then he/she must have proper credentials. They have to register by their email id and password. These credentials are only accessible to the administrator only.

Location Page

After entering the valid credentials, the login page is directed to location page. Here the administrator can add an employee, delete and can update the details of any employee.

Employee Details

The administrator has to add an employee if any new employee had joined. If the employee has to resign the job and wants to leave the college then the administrator has to delete that particular employee details. If the employee got salary increment or promotion for a higher position then the administrator has to update the details of that employee.

Attendance Directory

After doing the employee details there is an option ”CHOOSE FILE” in that frame such that if administrator selects that option it navigates directly to the attendance excel sheet that is generated from biometric machine. If the administrator selects the file and opens it then he/she has to push that file into the database by selecting the option “PUSH DATABASE”.

Salary Information

This module contains the information about the salary that each employee gets by the end of month. It contains the fields like “gross, net, hra, oa, lop, tax, leaves taken”. Hra and la are fixed to each employee and by calculating all these fields and subtracting it from normal salary the remaining salary gets credited to the employee.

4. CONCLUSION

Based on all the diagrams we can design the required functionalities and the flow of data that is to be maintained between each of them. By doing all this we can maintain the application without any bugs and errors. All the diagrams that are developed show us the functionalities of the website

C H A P T E R - 5
IMPLEMENTATION AND RESULTS

1. INTRODUCTION

The Working in this project is simple and easily understandable. The admin is provided with default user id and password. He can login to the portal and can perform his operations. Admin will be able to add/update/delete Employee details.

After opening the page using proper credentials then he/she has to push the attendance to the database using "push into database". If the admin selects that option then it will go to another frame that contains the option "choose file, update, generate". After selecting that it navigates directly to the attendance excel sheet which is generated from biometric machine. If the admin select update button it updates the attendance data in the database. By selecting the "generate" button it will go to next frame and displays that successfully generated the Employee details. In that frame there will be two options that contains "view results" and "generate results". If the admin selects "generate results" then it will generate the salary slip of each employee.

2. IMPLEMENTATION OF KEY FUNCTIONS

The Employee's attendance data is stored in Database and when the admin logins to the system the login credentials are checked, and home page of the employee details will be displayed if they are correct. Then the admin fetch the attendance details of each employee from excel sheet and push the updated attendance into the database. By analyzing the salary details of each employee, the admin select generate results that will generate the salary slip of each employee. If the admin choose view results it will display the salary slip of each employee in the database.

3. METHOD OF IMPLEMENTATION

```
import tkinter as tk
import mysql.connector
from tkinter import filedialog
import xlrd
import os
from os import listdir
import smtplib
```

```
c n x                                    =
mysql.connector.connect(user='root',passw
ord='1234',auth_plugin='mysql_native_pas
sword',host='127.0.0.1',database="sai")
mycursor=cnx.cursor()
#back function for page2
def back_function():
   frame2.pack_forget()
   entry.delete(0, tk.END)
   entry1.delete(0, tk.END)
   frame.pack()

#back funtion for page3 of add button
def aback_function():
   aframe3.pack_forget()
   frame2.pack()

#back funtion for remove employee button
def rback_function():
   rframe3.pack_forget()
   frame2.pack()

#back funtion for view emplyee button
def vback_funtion():
   vframe3.pack_forget()
   frame2.pack()
# funtion for update back button
def Uback_funtion():
   Uframe.pack_forget()
   frame2.pack()

#back funtion for generate back
def gbBack():
   Gframe.pack_forget()
   Uframe.pack()

#funtion for adding employee
def AddEmp():

   global counta
   if(counta==0):
      frame2.pack_forget()

label=tk.Label(master=aframe3,text="you
can add know",fg="black")
      label.pack()
                                    button1 =
tk.Button(master=aframe3,text="back",fg=
"black",command=aback_function)
      button1.pack()
      aframe3.pack()
      counta+=1
   else:
      frame2.pack_forget()
      aframe3.pack()



def mailData():
         mycursor.execute("select * from
employee")
```

```
    resultset=mycursor.fetchall()
    for i in resultset:
                                                    w i t h
smtplib.SMTP('smtp.gmail.com',587)   as
smtp:
            smtp.ehlo()
            smtp.starttls()
            smtp.ehlo()

smtp.login("saiscommercial@gmail.com","
Saiscommercial@955")
            sub="monthly salary slip"
                        msg=f'Subject  :  {sub}  \n  \n
'+"hra,oa,gross,net,it,pf,pt"+str(i)

smtp.sendmail("saiscommercial@gmail.co
m",str(i['id'])+"@mits.ac.in",msg)






#funtion for removing employee

def RemEmp():
    global countr
    if(countr==0):
        frame2.pack_forget()

label=tk.Label(master=rframe3,text="you
can remove know",fg="black")
        label.pack()
                                        b u t t o n r   =
tk.Button(master=rframe3,text="back",fg=
"black",command=rback_function)
        buttonr.pack()
        rframe3.pack()
        countr+=1
    else:
        frame2.pack_forget()
        rframe3.pack()

###this is upload method that traverse all
xlfiles and puts data to employee_personnel
def uploadData():

prolabel=tk.Label(master=Uframe,text="R
esults processig",fg="black")
    prolabel.grid(row=3,column=3)
    l1=[]
    for i in range(0,700):
        l1.append("")
    file_list=os.listdir(path)
    m=0
    st=""
    for j in file_list:
            wb = xlrd.open_workbook(str(path)
+"\\"+str(j))
        sheet = wb.sheet_by_index(0)
        global k
        k=sheet.nrows
    global inputtxt
                    inputtxt=
tk.Text(master=Uframe,height=30,width=1
20)
    inputtxt.grid(row=5,column=2)
    for i in range(9,k):
        if(len(sheet.cell_value(i,1))==8 and
sheet.cell_value(i,1)!="Security"):
                #mycursor.execute("select
attendance from employee where
id='{}'".format(sheet.cell_value(i,1)))
                        if("PR" in
str(sheet.cell_value(i,12)) or "PR" in
str(sheet.cell_value(i,13))):
            #print(i)
            l1[i-9]+='1'
            #up=str(mycursor.fetchone()[0])
+'1'
               #mycursor.execute("update
employee set attendance='{}' where
id='{}'".format(up,sheet.cell_value(i,1)))
        else:
            #print(i)
            l1[i-9]+='0'
            #up=str(mycursor.fetchone()[0])
+'0'
               #mycursor.execute("update
employee set attendance='{}' where
id='{}'".format(up,sheet.cell_value(i,1)))


    m+=1
```

```
            st+=str(m)+"\t wait ur file"+str(j)
+"just now uploaded\n"
      inputtxt.insert(tk.END,str(m)+st)
              prolabel.config(text="finished
uploading")
   for i in range(9,k):
          mycursor.execute("update employee
set attendance='{}' where
id='{}'".format("".join(l1[i-9]),sheet.cell_va
lue(i,1)))
      cnx.commit()




###function that generates slip
def generateData():
   for i in range(9,k):
            if(len(sheet.cell_value(i,1))==8 and
sheet.cell_value(i,1)!="Security"):
          mycursor.execute("select attendance
from employee where
id='{}'".format(sheet.cell_value(i,1)))
           attendance=str(mycursor.fetchone()
[0])
          days=len(attendance)
          no_of_atd=str(attendance)
          no_of_atd=no_of_atd.count('1')

no_of_abd=str(attendance).count('0')
          mycursor.execute("select basic from
employee_personnel where
id='{}'".format(sheet.cell_value(i,1)))
        basic=float(mycursor.fetchone()[0])
        hra=2000
        oa=2000
            mycursor.execute("select holidas
from holidays2021 where
month='{}'".format('oct'))
        holidas=int(mycursor.fetchone()[0])
        workingdays=days-holidas
         mycursor.execute("select leaves_left
from employee_personnel where
id='{}'".format(sheet.cell_value(i,1)))
          leaves_left=int(mycursor.fetchone()
[0])
              if(workingdays>no_of_atd and
leaves_left-(workingdays-no_of_atd)>0):
                   leaves_left-=workingdays-
no_of_atd
                   mycursor.execute("update
employee_personnel set leaves_left='{}'
where
id='{}'".format(leaves_left,sheet.cell_value(
i,1)))
           cnx.commit()
           lop=0.00
            elif(workingdays>no_of_atd and
leaves_left==0):
                   lop=basic*((workingdays-
no_of_atd)/workingdays)
                     basic=basic*(no_of_atd/
workingdays)
```

```
        else:
           lop=0.00
        gross=basic+hra+oa
        it=basic*0.12
        pf=basic*0.10
        pt=basic*0.05
        net=gross-it-pf-pt
        mycursor.execute("update employee
s e t
no_of_atd='{}',gross='{}',net='{}',hra='{}',o
a='{}',it='{}',pf='{}',pt='{}',lop='{}' where
id='{}'".format(no_of_atd,gross,net,hra,oa,i
t,pf,pt,lop,sheet.cell_value(i,1)))
        cnx.commit()
          inputtxt.insert(tk.END,"the salary
details have been caluculated\n")



#function that lets user to select the
directory
def getDir():
   global path
                                      path=
filedialog.askdirectory(initialdir="/",
title="Select file")

plabel=tk.Label(master=Uframe,text="("+
path+")",fg="black")
   plabel.grid(row=2,column=3)
```

```
#funtion for viewing employee details
def ViewEmp():
   global countv
   if(countv==0):
      frame2.pack_forget()

label=tk.Label(master=vframe3,text="you
can View know",fg="black")
      label.pack()
                                    b u t t o n v   =
tk.Button(master=vframe3,text="back",fg=
"black",command=vback_funtion)
      buttonv.pack()
      vframe3.pack()
      countv+=1
   else:
      frame2.pack_forget()
      vframe3.pack()

# funtion for update the data base with new
xl file
def Update():
   global Ucount
   if(Ucount==0):
      frame2.pack_forget()

ubutton=tk.Button(master=Uframe,text="
ChooseDir",fg="black",command=getDir)
       ubutton.grid(row = 2, column = 2,padx
= 10,pady=10,ipadx=100,ipady=5)
```

```
gbutton=tk.Button(master=Uframe,text="
Upload",fg="black",command=uploadDat
a)
	gbutton.grid(row = 3, column = 2,padx
= 10,pady=10,ipadx=100,ipady=5)
					b u t t o n u   =
tk.Button(master=Uframe,text="back",fg=
"black",command=Uback_funtion)
	buttonu.grid(row = 8, column = 2,padx
= 10,pady=10,ipadx=100,ipady=5)
					b u t t o n u   =
tk.Button(master=Uframe,text="generate",
fg="black",command=generateData)
	buttonu.grid(row = 6, column = 2,padx
= 10,pady=10,ipadx=100,ipady=5)

label2=tk.Label(master=Uframe,text="clic
k here to generate sal details",fg="black")
	label2.grid(row = 6, column = 3,padx =
10,pady=10)
					b u t t o n u   =
tk.Button(master=Uframe,text="send_to_
mail",fg="black",command=mailData)
	buttonu.grid(row = 7, column = 2,padx
= 10,pady=10,ipadx=100,ipady=5)

label2=tk.Label(master=Uframe,text="clic
k here to send mail",fg="black")
	label2.grid(row = 7, column = 3,padx =
10,pady=10)
```

```
        Uframe.pack()


#cbutton=tk.Button(master=Uframe,text="
ChooseDir",fg="black",command=getDir)
            #cbutton.grid(row = 3, column =
2,padx = 10,pady=10,ipadx=100,ipady=5)
        Ucount+=1
    else:
        frame2.pack_forget()
        Uframe.pack()
#funtion for generate page
def Genarate_funtion():
    global gcount
    if(gcount==0):
        Uframe.pack_forget()

dbutton=tk.Button(master=Gframe,text="
ViewDB",fg="black")
        dbutton.grid(row = 1, column = 0,padx
= 10,pady=10,ipadx=100,ipady=5)

gbbutton=tk.Button(master=Gframe,text="
back",fg="black",command=gbBack)
            gbbutton.grid(row = 2, column =
0,padx = 10,pady=10,ipadx=100,ipady=5)
        Gframe.pack()
        gcount+=1
    else:
        Uframe.pack_forget()
        Gframe.pack()








#funtion for displaying page2
def function():
    global count
    if(count==0):
        uname=entry.get()
        password=entry1.get()
        if(uname=="" or password==""):

flabel=tk.Label(master=frame,text="details
cant be empty",fg="black")
            flabel.grid(row=3,column = 1,padx =
10,pady=10)



        else:
                              mycursor.execute("select
id,password from admin where id='{}' and
password='{}'".format(uname,password))
            details=mycursor.fetchone()
            if(not details):

flabel=tk.Label(master=frame,text="invali
d authentication",fg="black")
                       flabel.grid(row=3,column =
1,padx = 10,pady=10)

                elif(str(details[0])==uname and
str(details[1])==password):
            frame.pack_forget()

label2=tk.Label(master=frame2,text="Hai
"+uname,fg="black")
                label2.grid(row = 0, column =
1,padx = 10,pady=10)
                                    add_emp =
tk.Button(master=frame2,text="Add_Emp
", fg="black",command=AddEmp)
             add_emp.grid(row = 1, column =
0,padx = 10,pady=10,ipadx=100,ipady=5)
                                    rem_emp =
tk.Button(master=frame2,text="Rem_Emp
", fg="black",command=RemEmp)
             rem_emp.grid(row = 2, column =
0,padx = 10,pady=10,ipadx=100,ipady=5)
                                   view_emp =
tk.Button(master=frame2,text="View_Emp
", fg="black",command=ViewEmp)
             view_emp.grid(row = 1, column =
2,padx = 10,pady=10,ipadx=100,ipady=5)

buttonU=tk.Button(master=frame2,text="u
pdate File",fg="black",command=Update)
               buttonU.grid(row = 2, column =
2,padx = 10,pady=10,ipadx=100,ipady=5)
            frame2.pack()
```

```
                button1 =
tk.Button(master=frame2,text="back",fg=
"black",command=back_function)
            button1.grid(row = 3, column =
1,padx = 10,pady=10,ipadx=100,ipady=5)
            count+=1

    else:
        frame.pack_forget()
        frame2.pack()
#creating main window and frames for all
required sub pages

window = tk.Tk()

frame = tk.Frame()
frame2 = tk.Frame()
aframe3=tk.Frame()
rframe3=tk.Frame()
vframe3=tk.Frame()
Uframe=tk.Frame()
Gframe=tk.Frame()

label = tk.Label(master=frame,text="User
Name", fg="black")
label.grid(row = 0, column = 0,padx =
10,pady=10)

entry = tk.Entry(master=frame)
entry.grid(row = 0, column =
1,ipadx=50,ipady=5)
```

```
label1 = tk.Label(master=frame,text="Password", fg="black")
label1.grid(row = 1, column = 0,padx = 10,pady=10)


entry1 = tk.Entry(master=frame)
entry1.grid(row = 1, column = 1,ipadx=50,ipady=5)



count=0
counta=0
countr=0
countv=0
Ucount=0
gcount=0



button = tk.Button(master=frame,text="login",fg="black",command=function) ###this one login button goes to function
```

```
button.grid(row=2,column = 1,padx =
10,pady=10)
frame.pack()
window.mainloop()
cnx.close()
```

4. OUTPUT SCREENS AND RESULT ANALYSIS

Figure 5.4.1: Login Page

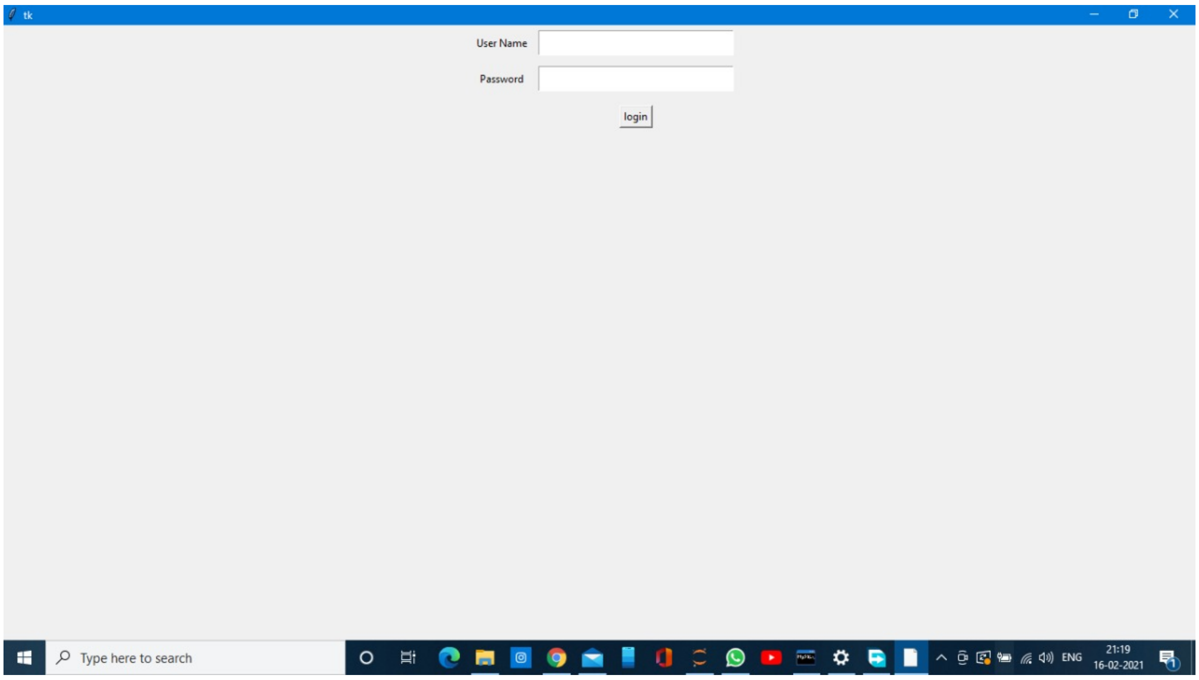


Figure 5.4.1: Admin Login Page

Home Page:

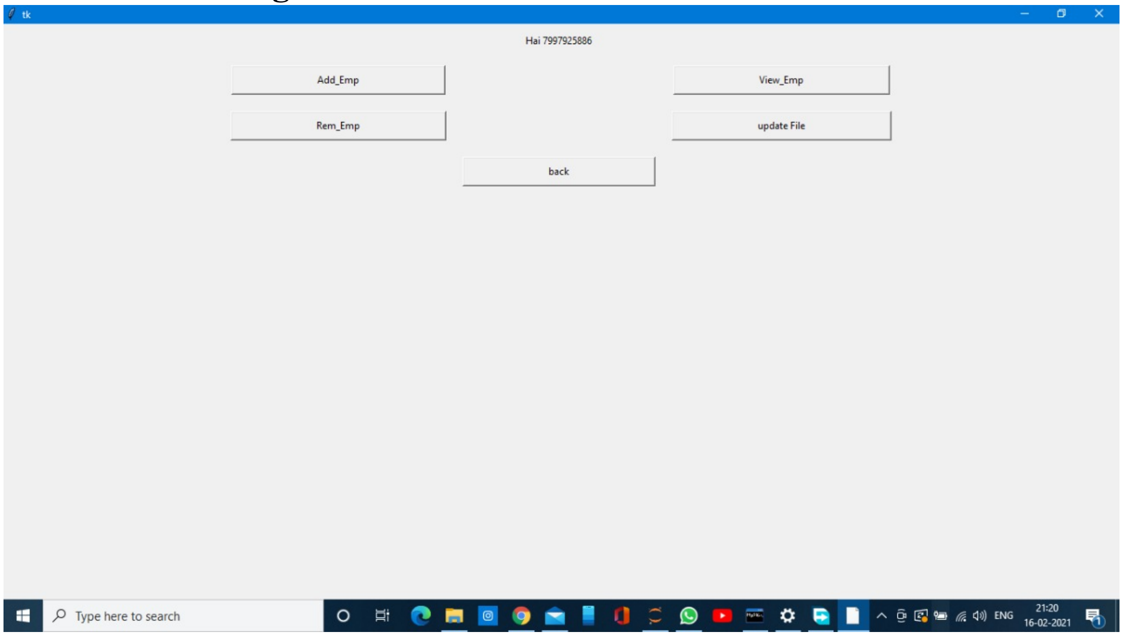


Figure 5.4.2:Home page

Update Employee page:

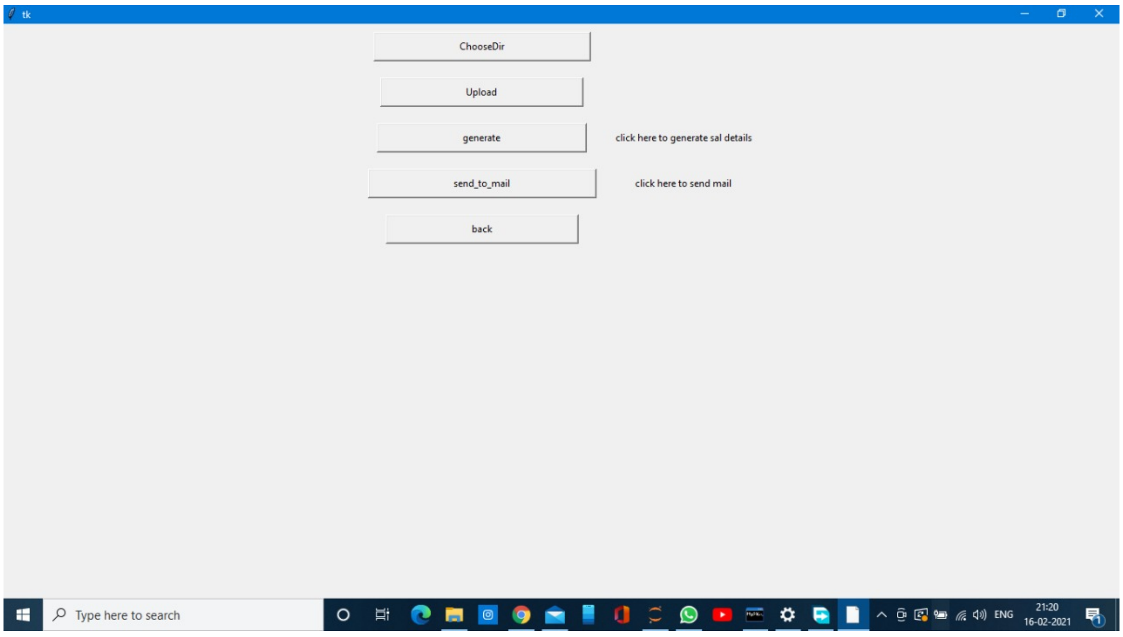
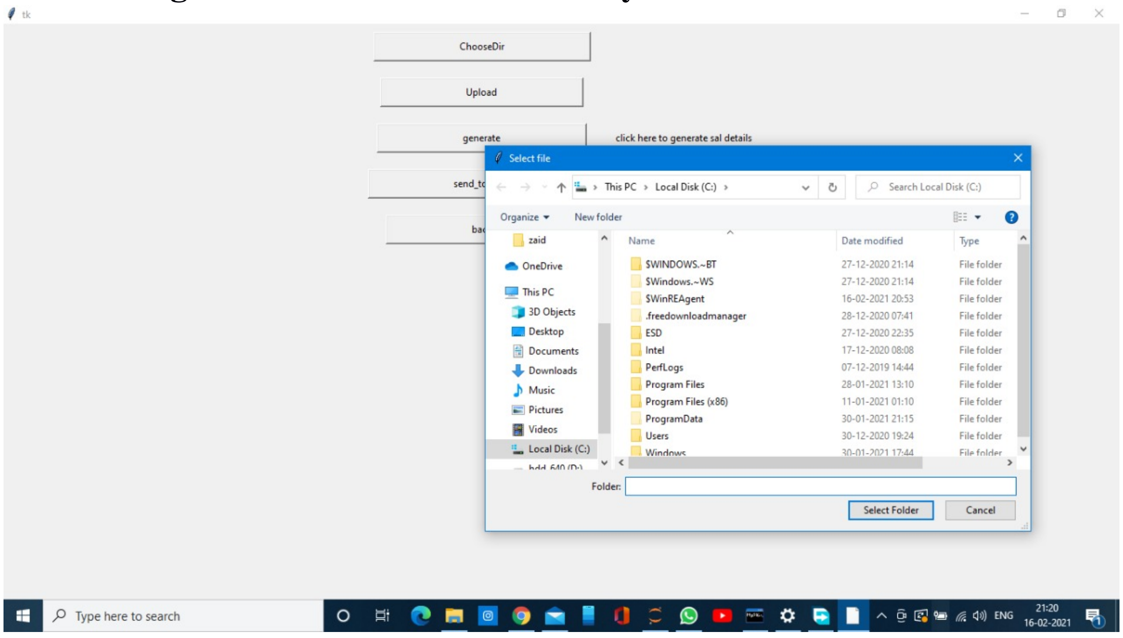


Figure 5.4.3 Update Employee Page

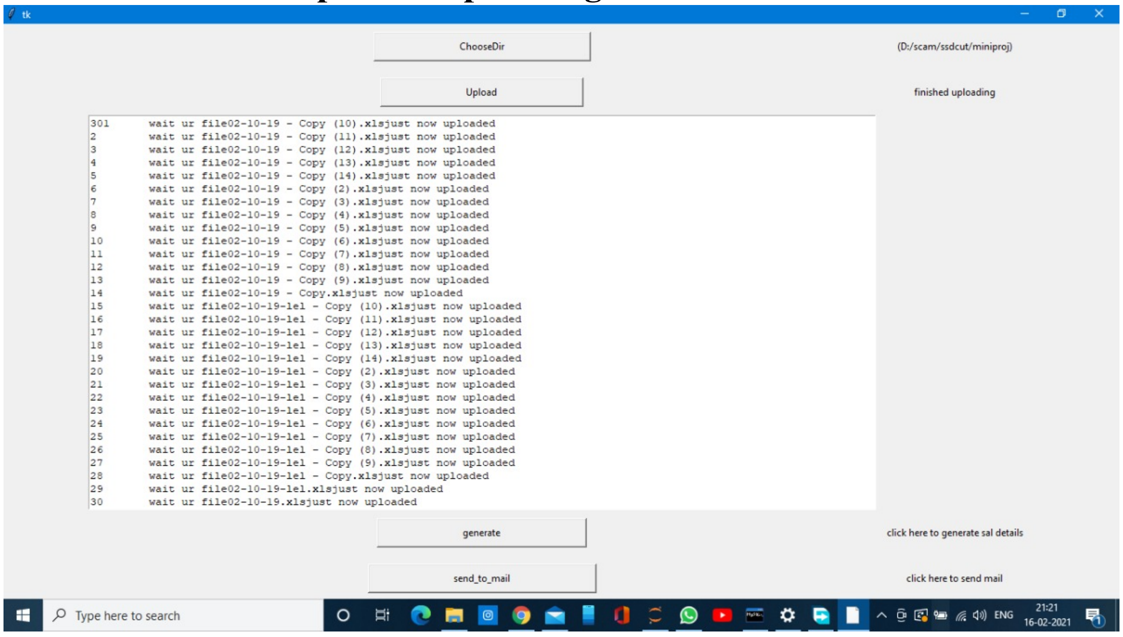
Choosing Attendance file in Directory



Name	Date modified	Type
$WINDOWS.~BT	27-12-2020 21:14	File folder
$Windows.~WS	27-12-2020 21:14	File folder
$WinREAgent	16-02-2021 20:53	File folder
.freedownloadmanager	28-12-2020 07:41	File folder
ESD	27-12-2020 22:35	File folder
Intel	17-12-2020 08:08	File folder
PerfLogs	07-12-2019 14:44	File folder
Program Files	28-01-2021 13:10	File folder
Program Files (x86)	11-01-2021 01:10	File folder
ProgramData	30-01-2021 21:15	File folder
Users	30-12-2020 19:24	File folder
Windows	30-01-2021 17:44	File folder

Figure 5.4.4: Choosing Attendance file in Directory

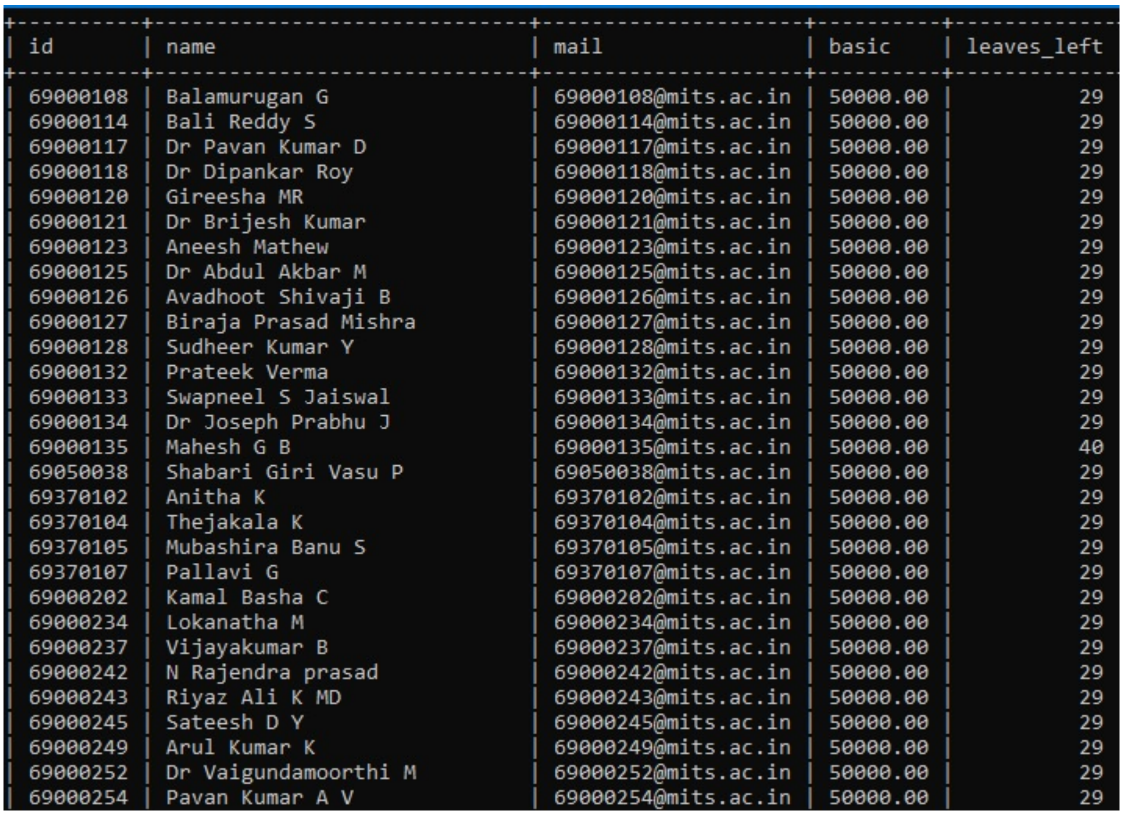
Module Output for Uploading Attendance data:



```
301      wait ur file02-10-19 - Copy (10).xlsjust now uploaded
2        wait ur file02-10-19 - Copy (11).xlsjust now uploaded
3        wait ur file02-10-19 - Copy (12).xlsjust now uploaded
4        wait ur file02-10-19 - Copy (13).xlsjust now uploaded
5        wait ur file02-10-19 - Copy (14).xlsjust now uploaded
6        wait ur file02-10-19 - Copy (2).xlsjust now uploaded
7        wait ur file02-10-19 - Copy (3).xlsjust now uploaded
8        wait ur file02-10-19 - Copy (4).xlsjust now uploaded
9        wait ur file02-10-19 - Copy (5).xlsjust now uploaded
10       wait ur file02-10-19 - Copy (6).xlsjust now uploaded
11       wait ur file02-10-19 - Copy (7).xlsjust now uploaded
12       wait ur file02-10-19 - Copy (8).xlsjust now uploaded
13       wait ur file02-10-19 - Copy (9).xlsjust now uploaded
14       wait ur file02-10-19 - Copy.xlsjust now uploaded
15       wait ur file02-10-19-lel - Copy (10).xlsjust now uploaded
16       wait ur file02-10-19-lel - Copy (11).xlsjust now uploaded
17       wait ur file02-10-19-lel - Copy (12).xlsjust now uploaded
18       wait ur file02-10-19-lel - Copy (13).xlsjust now uploaded
19       wait ur file02-10-19-lel - Copy (14).xlsjust now uploaded
20       wait ur file02-10-19-lel - Copy (2).xlsjust now uploaded
21       wait ur file02-10-19-lel - Copy (3).xlsjust now uploaded
22       wait ur file02-10-19-lel - Copy (4).xlsjust now uploaded
23       wait ur file02-10-19-lel - Copy (5).xlsjust now uploaded
24       wait ur file02-10-19-lel - Copy (6).xlsjust now uploaded
25       wait ur file02-10-19-lel - Copy (7).xlsjust now uploaded
26       wait ur file02-10-19-lel - Copy (8).xlsjust now uploaded
27       wait ur file02-10-19-lel - Copy (9).xlsjust now uploaded
28       wait ur file02-10-19-lel - Copy.xlsjust now uploaded
29       wait ur file02-10-19-lel.xlsjust now uploaded
30       wait ur file02-10-19.xlsjust now uploaded
```

Figure 5.4.5 Module output for uploading Attendance Data

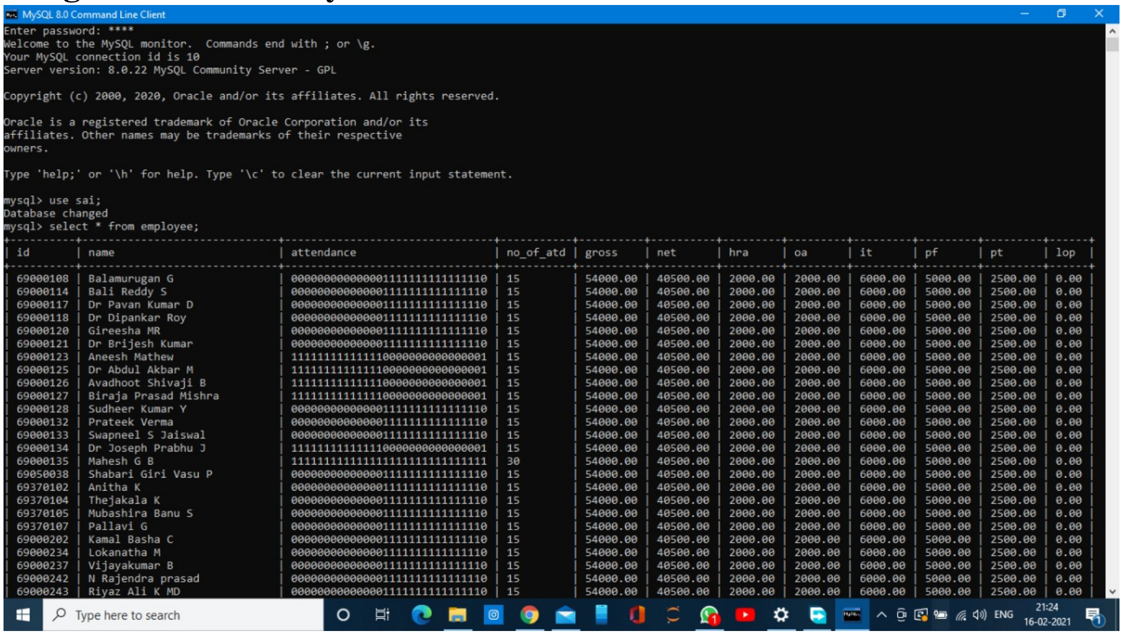
Employee Details Database



```
+----------+----------------------------+---------------------+----------+------------
| id       | name                       | mail                | basic    | leaves_left
+----------+----------------------------+---------------------+----------+------------
| 69000108 | Balamurugan G              | 69000108@mits.ac.in | 50000.00 |          29
| 69000114 | Bali Reddy S               | 69000114@mits.ac.in | 50000.00 |          29
| 69000117 | Dr Pavan Kumar D           | 69000117@mits.ac.in | 50000.00 |          29
| 69000118 | Dr Dipankar Roy            | 69000118@mits.ac.in | 50000.00 |          29
| 69000120 | Gireesha MR                | 69000120@mits.ac.in | 50000.00 |          29
| 69000121 | Dr Brijesh Kumar           | 69000121@mits.ac.in | 50000.00 |          29
| 69000123 | Aneesh Mathew              | 69000123@mits.ac.in | 50000.00 |          29
| 69000125 | Dr Abdul Akbar M           | 69000125@mits.ac.in | 50000.00 |          29
| 69000126 | Avadhoot Shivaji B         | 69000126@mits.ac.in | 50000.00 |          29
| 69000127 | Biraja Prasad Mishra       | 69000127@mits.ac.in | 50000.00 |          29
| 69000128 | Sudheer Kumar Y            | 69000128@mits.ac.in | 50000.00 |          29
| 69000132 | Prateek Verma              | 69000132@mits.ac.in | 50000.00 |          29
| 69000133 | Swapneel S Jaiswal         | 69000133@mits.ac.in | 50000.00 |          29
| 69000134 | Dr Joseph Prabhu J         | 69000134@mits.ac.in | 50000.00 |          29
| 69000135 | Mahesh G B                 | 69000135@mits.ac.in | 50000.00 |          40
| 69050038 | Shabari Giri Vasu P        | 69050038@mits.ac.in | 50000.00 |          29
| 69370102 | Anitha K                   | 69370102@mits.ac.in | 50000.00 |          29
| 69370104 | Thejakala K                | 69370104@mits.ac.in | 50000.00 |          29
| 69370105 | Mubashira Banu S           | 69370105@mits.ac.in | 50000.00 |          29
| 69370107 | Pallavi G                  | 69370107@mits.ac.in | 50000.00 |          29
| 69000202 | Kamal Basha C              | 69000202@mits.ac.in | 50000.00 |          29
| 69000234 | Lokanatha M                | 69000234@mits.ac.in | 50000.00 |          29
| 69000237 | Vijayakumar B              | 69000237@mits.ac.in | 50000.00 |          29
| 69000242 | N Rajendra prasad          | 69000242@mits.ac.in | 50000.00 |          29
| 69000243 | Riyaz Ali K MD             | 69000243@mits.ac.in | 50000.00 |          29
| 69000245 | Sateesh D Y                | 69000245@mits.ac.in | 50000.00 |          29
| 69000249 | Arul Kumar K               | 69000249@mits.ac.in | 50000.00 |          29
| 69000252 | Dr Vaigundamoorthi M       | 69000252@mits.ac.in | 50000.00 |          29
| 69000254 | Pavan Kumar A V            | 69000254@mits.ac.in | 50000.00 |          29
```

Figure 5.4.6: Employee Details Database

Figure 5.4.7: Salary and Attendance Details Database



```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use sai;
Database changed
mysql> select * from employee;
+----------+----------------------------+--------------------------------+-----------+----------+----------+---------+---------+---------+---------+---------+------+
| id       | name                       | attendance                     | no_of_atd | gross    | net      | hra     | oa      | it      | pf      | pt      | lop  |
+----------+----------------------------+--------------------------------+-----------+----------+----------+---------+---------+---------+---------+---------+------+
| 69000108 | Balamurugan G              | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000114 | Bali Reddy S               | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000117 | Dr Pavan Kumar D           | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000118 | Dr Dipankar Roy            | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000120 | Gireesha MR                | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000121 | Dr Brijesh Kumar           | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000123 | Aneesh Mathew              | 111111111111111000000000000001 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000125 | Dr Abdul Akbar M           | 111111111111111000000000000001 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000126 | Avadhoot Shivaji B         | 111111111111111000000000000001 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000127 | Biraja Prasad Mishra       | 111111111111111000000000000001 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000128 | Sudheer Kumar Y            | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000132 | Prateek Verma              | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000133 | Swapneel S Jaiswal         | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000134 | Dr Joseph Prabhu J         | 111111111111111000000000000001 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000135 | Mahesh G B                 | 111111111111111111111111111111 | 30        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69050038 | Shabari Giri Vasu P        | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69370102 | Anitha K                   | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69370104 | Thejakala K                | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69370105 | Mubashira Banu S           | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69370107 | Pallavi G                  | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000202 | Kamal Basha C              | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000234 | Lokanatha M                | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000237 | Vijayakumar B              | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000242 | N Rajendra prasad          | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
| 69000243 | Riyaz Ali K MD             | 000000000000000111111111111110 | 15        | 54000.00 | 40500.00 | 2000.00 | 2000.00 | 6000.00 | 5000.00 | 2500.00 | 0.00 |
```

Figure 5.4.7: Salary and Attendance Details Database

5. CONCLUSION

Hence from the above proposed model we have generated the salary slip of an employee based on the monthly attendance and send it to them through E-mail which will be done automatically.

C H A P T E R - 6

TESTING AND VALIDATION

1. INTODUCTION

INTRODUCTION TO TESTING

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development

UNIT TESTING

The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. Unit Testing is a level of software testing where individual units/components of software are tested. Typically, the unit test will establish some sort of artificial environment and then invoke methods in the unit being tested. It then checks the results returned against some known value. When the units are assembled, we can use the same tests to test the system as a hole. It usually has one or a few inputs and usually a single output.

FUNCTIONAL TESTING

Functional Testing is a testing technique that is used to test the features functionality of the system or Software should cover all the scenarios including failure paths and boundary cases. For example, validation of contents of database.

INTEGRATION TESTING

Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated. Similarly, every unit is integrated after the testing of every single unit is done individually.

SYSTEM TESTING

System testing of software or hardware is the testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. The hardware and the software units are tested separately and then tested together to check if the desired results are obtained.

PERFORMANCE TESTING

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

2. DESIGN OF TEST CASES AND SCENARIOS

Testing of RFID reader

Number of tests carried	Tags read Successfully	Tags failed to read
5	4	1

Testing of Zigbee transmission

Number of tests carried	Sent	Received	Failed
5	5	4	1

3. VALIDATION

The following test case scenarios were used in the integrated system testing to prove the working of the developed system.

a) Login of admin.
b) Add, update and delete the details of the employee.
c) Fetch the monthly attendance details from excel sheet and update it in the Database.
d) Calculate the monthly salary by update it in the database.
e) Generate the salary slip and send it to the employees through E-mail Automatically.

4. CONCLUSION

The proposed model is easy to use. The admin can login into the system and can modify the details of any employee. He/she fetches the attendance details from excel sheet and updates it in the database.

CHAPTER-7
CONCLUSION

7.1 CONCLUSION

Hence from the above proposed method we have described the Employee’s salary generation slip working. With this type of system, the admin can easily send the salary slip to the employees through email which will be done automatically.

The proposed model is easy to use. The admin is provided with default user id and password. He can login to the portal and can perform his operations. Admin will be able to add/update/delete details of the employee.

The admin fetches the attendance in the system that has generated in excel sheet from biometric machine and updates it in the database. By analyzing all the leaves and holidays that the employee has taken in the month our application calculates the salary information and deduct the money from the salary and generates the salary slip. When the admin clicks “view results” it will display salary slip of every employee.

CHAPTER-8
REFERENCES

REFERENCES

1. . http//www.ijetae.com/files/Volume4Issue4/IJE TAE_0414_140.

2. Schmitt, Bob (10 October 2014). & quot; Collection Management Systems". Retrieved 15 November 2015.

3. Carpinone, Elana C. (May 2010), Museum Collection Management systems: One Size Does NOT Fit All, p. 26. Retrieved 5 December 2015.

4. http://www.tangedco.gov.in/linkpdf/Agri%20For m.pdf: The pdf is a form to apply for agricultural loan. This gives us the idea about attributes required in database "Agricultural loans".

5. http://ceoharyana.nic.in/?module=pages&pageid =42: The link tells us about the requirements for Voter ID Generation. This information can be used to maintain the database.

6. A Java API for the description of large complex networks under the object-oriented paradigm A. Ni~no1;y, C. Mu~noz-Caro1, S. Reyes1 and M. Castillo1 1 SciCom Research Group. Escuela Superior de Inform_atica de Ciudad Real,Universidad de Castilla-La Mancha. Paseo de la Universidad 4. 13004 Ciudad Real, Spain.

7. Compiling and Optimizing Java 8 Programs for GPU Execution Kazuaki Ishizaki IBM Research– Tokyo ,Akihiro Hayashi Rice University ,Gita Koblents IBM Canada, Vivek Sarkar Rice University