Topic : Office automation system

Group no :15

Campus : Kandy

Submission Date:

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21264184	R.V.A.N.S Wijerathne	0772576120
IT21270710	W.A.I.A Bandara	0756132962
IT21288630	R.G.G.P Kulathunga	0767671702
IT21323584	H.M.D.N.S Heenkenda	0714059426
IT21244902	C.J Widanage	0761866252

Exercise 01

I. User Requirements

1. To gain access to the system, The employee must first register with the system by entering an email address and a password.
2. Employee is one of 3 users of this system. An employee can mark their attendance everyday by choosing the date and time.
3. Department manager is another user of the system. The relevant department manager can view the employee attendance sheet.
4. A registered employee can request many resources. They can request a resource by selecting the resource id.
5. When a resource is requested, the admin will be notified. Then admin will pass the request to the relevant department manager. Admin is also a user of this system.
6. After the manager's inspection, approval will be granted and he will send a confirmation email.
7. The employee can give feedback to the admin.
8. The admin can view zero or many feedback.
9. Manager can organize events and contact employees through the system.
10. The manager should update the progress of the events regularly.
11. The admin can add or remove accounts.

Nouns

1. system
2. employee
3. email
4. address
5. password
6. attendance
7. date
8. time
9. department
10. manager
11. attendance
12. sheet
13. registered employee
14. resource
15. item
16. admin
17. request
18. confirmation email
19. feedback

20. progress
21. events
22. database
23. accounts

Verbs

Noun	Verb
Visitor(user)	Gain Register Entering
Registered employee	mark choosing request selecting give
Manager	Edit inspection granted sent organize contact update
Admin	notified passed maintains updates add remove

Classes

1. User
2. Employee
3. Admin
4. Manager
5. Resource
6. Attendance
7. Feedback
8. Event

Exercise 02

Class Name: User	
Responsibilities	Collaboration
Create account	Employee Admin Manager

Class Name: Employee (registered)	
Responsibilities	Collaboration
Log in Request resource Mark attendance Give feedback	Resource Attendance Feedback

Class Name: Resource	
Responsibilities	Collaboration
Add resource Update resource	

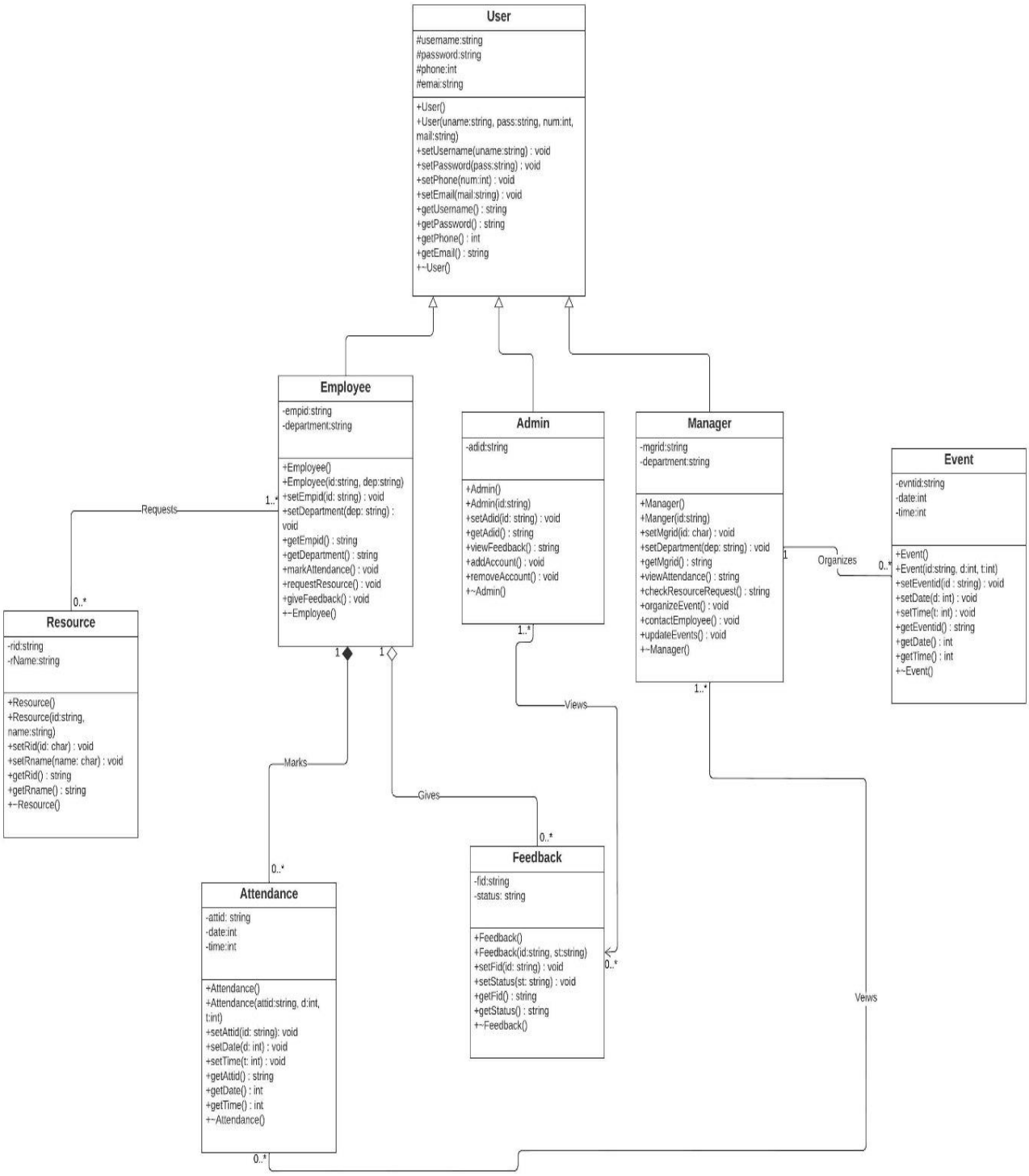
Class Name: Attendance	
Responsibilities	Collaboration
View attendance	Employee
Add attendance	Manager
Update attendance	Manager

Class Name: Manager	
Responsibilities	Collaboration
Organize events Edit attendance Approve request	Event Attendance Resource

Class Name: Feedback	
Responsibilities	Collaboration
Store feedback	

Class Name: Event	
Responsibilities	Collaboration
View event Set date and time	

Class Name: Admin	
Responsibilities	Collaboration
View feedback	Feedback
Add account	Employee
Remove account	Employee


User
#username:string
#password:string
#phone:int
#emai:string
+User()
+User(uname:string, pass:string, num:int, mail:string)
+setUsername(uname:string) : void
+setPassword(pass:string) : void
+setPhone(num:int) : void
+setEmail(mail:string) : void
+getUsername() : string
+getPassword() : string
+getPhone() : int
+getEmail() : string
+~User()
Employee
-empid:string
-department:string
+Employee()
+Employee(id:string, dep:string)
+setEmpid(id: string) : void
+setDepartment(dep: string) : void
+getEmpid() : string
+getDepartment() : string
+markAttendance() : void
+requestResource() : void
+giveFeedback() : void
+~Employee()
Admin
-adid:string
+Admin()
+Admin(id:string)
+setAdid(id: string) : void
+getAdid() : string
+viewFeedback() : string
+addAccount() : void
+removeAccount() : void
+~Admin()
Manager
-mgrid:string
-department:string
+Manager()
+Manger(id:string)
+setMgrid(id: char) : void
+setDepartment(dep: string) : void
+getMgrid() : string
+viewAttendance() : string
+checkResourceRequest() : string
+organizeEvent() : void
+contactEmployee() : void
+updateEvents() : void
+~Manager()
Event
-evntid:string
-date:int
-time:int
+Event()
+Event(id:string, d:int, t:int)
+setEventid(id : string) : void
+setDate(d: int) : void
+setTime(t: int) : void
+getEventid() : string
+getDate() : int
+getTime() : int
+~Event()
Resource
-rid:string
-rName:string
+Resource()
+Resource(id:string, name:string)
+setRid(id: char) : void
+setRname(name: char) : void
+getRid() : string
+getRname() : string
+~Resource()
Attendance
-attid: string
-date:int
-time:int
+Attendance()
+Attendance(attid:string, d:int, t:int)
+setAttid(id: string): void
+setDate(d: int) : void
+setTime(t: int) : void
+getAttid() : string
+getDate() : int
+getTime() : int
+~Attendance()
Feedback
-fid:string
-status: string
+Feedback()
+Feedback(id:string, st:string)
+setFid(id: string) : void
+setStatus(st: string) : void
+getFid() : string
+getStatus() : string
+~Feedback()
Requests
1..*
0..*
Organizes
1
0..*
Views
1..*
Marks
Gives
1
1
0..*
0..*
Views
0..*
0..*

User.h

```
//contributor-Wijerathne.R.V.A.N.S – IT21264184
#pragma once
class User
{
private:
	int phone;
	string username;
	string email;
	string password;
public:
	User();
	void setUsername(string uname);
	void setPassword(string pass);
	void setEmail(string mail);
	void setPhone(string num);
	string getUsername();
	string getPassword();
	string getEmail();
	string getPhone();
	virtual void DisplayUserdetails();
    ~User();
};
```

User.cpp

```
//contributor-Wijerathne.R.V.A.N.S – IT21264184
#include <iostream>
#include "User.h"
#include <string.h>
using namespace std;

User::User(void)
{
}


void User::setUsername(string uname)
{
	username = uname;
}
```

```
void User::setPassword(string pass)
{
	Password = pass;
}

void User::setEmail(string mail)
{
	email = mail;
}

void User::setPhone(const int num)
{
	phone = num;
}


string User::getUsername()
{
	return username;
}

string User::getPassword()
{
	return password;
}

string User::getEmail()
{
	return email;
}

int* User::getPhone()
{
	return phone;
}

void User::DisplayUserdetails()
{
	cout <<"Username : "<< username << endl;
	cout <<"Password : "<< password << endl;
	cout <<"Phone Number : "<< phone << endl;
	cout << "Email : "<< email << endl;
}

User::~User()
{
	cout<<"User deleted "<< endl;
}
```

Employee.h

```
//contributor-Wijerathne.R.V.A.N.S – IT21264184
#pragma once

class Employee
{
private:
	string empId;
	string department;

public:
	Employee();
	Employee(string Id, string Dep);
	void setEmpId(string Id);
	void setDepartment(string Dep[]);
	string getEmpId();
	string getDepartment;
	void markAttendance();
	void requestResource();
	void giveFeedback();
	~Employee(void);
};
```

Employee.cpp

```
//contributor-Wijerathne.R.V.A.N.S – IT21264184
#include <iostream>
#include <string.h>
#include "Employee.h"
using namespace std;

Employee::Employee(void)
{
	empId="";
	department="";
}

Employee::Employee(string Id, string Dep)
{
	empId= Id;
	department= Dep;
}
```

```
void Employee::setEmpId(string Id)
{
	empId= Id;
}

void Employee::setDepartment(string Dep)
{
	department= Dep;
}

string Employee::getEmpId()
{
	return empId;
}

string Employee::getDepartment()
{
	return department;
}

void Employee::markAttendance()
{
}

void Employee::requestResource()
{
}

void Employee::giveFeedback()
{
}

Employee::~Employee(void)
{
	cout << "Employee deleted" << endl;
}
```

Resources.h

```
//contributor - W.A.I.A.Bandara - IT21270710
#pragma once
class Resources
{
private:
string rid;
string rName;

public:
Resources();
Resources(string id, string name);
void setRid(char id);
void setRname(char name);
string getRid();
string getRname();
~Resources();
};
```

Resources.cpp

```
//contributor - W.A.I.A.Bandara - IT21270710
```

```cpp
#include <iostream>
#include "Resources.h"
#include <string>
using namespace std;

Resources::Resources() {

}
Resources::Resources(string id, string name) {
rid = id;
rName = name;
}
void Resources::setRid(char id) {
rid = id;
}
void Resources::setRname(char name) {
rName = name;
}
string Resources::getRid() {
return rid;
}
string Resources::getRname() {
return rName;
}
Resources::~Resources() {
cout << "Resource deleted" << endl;
}
```

Manager.h

```
//contributor - W.A.I.A.Bandara - IT21270710
#pragma once
class Manager:Attendance
{
private:
string mgrid;
string department;
  string Mname;

public:
Manager();
Manager(string id, string name, string dep) ;
void setMgrid(char id);
  void setMgname(string name);
void setDepartment(string dep);
  void getMgrid();
  void getMgname();
  void getDep();
string viewAttendance();
string checkResourceRequest();
void organizeEvent();
void contactEmployee();
void updateEvents();

~Manager();
```

Manager.cpp

```
//contributor - W.A.I.A.Bandara - IT21270710
#include <iostream>
#include "Manager.h"
#include "Attendance"
#include <string.h>
using namespace std;

Manager::Manager() {

}
Manager::Manager(string id, string name, string dep) {
mgrid = id;
  Mname = name
  department = dep;
}

void Manager::setMgname(string name){
  Mname = name;
}
void Manager::setMgrid(char id) {
mgrid = id;
}
void Manager::setDepartment(string dep) {
department = dep;
}
```

```
string Manager::getMgrid() {
return mgrid;
}
string Manager::getMgname() {
return Mname;
}
string Manager::getDep() {
return department;
}
string Attendance::viewAttendance() {

}
string Manager::checkResourceRequest() {

}

void Manager::organizeEvent() {

}
void Manager::contactEmployee(){
}

void Manager::updateEvents(){
}
Manager::~Manager(){
cout << "Manager deleted!" << endl;
```

Attendance. h

```
//Contributor- H. M. N. D. S. Heenekenda - IT21323584

#pragma once
class Attendance
{

private:
int date;
int time;
string attid;

public:
Attendance();
Attendance(string id);

void setAttid(string id);
void setDate();
void setTime();
string getAttid();
int getDate;
int getTime;

~Attendance();

};
```

Attendance. cpp

```
//Contributor- H. M. N. D. S. Heenekenda -IT21323584

#include <iostream>
#include<string>
#include "Attendance.h"
using namespace std;
```

```
Attendance::Attendance()
{
}

Attendance::Attendance(string id)
{
attid = id;
}

void Attendance::setAttid(string id)
{
    attid = id;
}

void Attendance::setDate(int d)
{
Date = d;
}

void Attendance::setTime(int t)
{
Time = t;
}

string Attendance::getAttid()
{
return attid;
}

string Attendance::getDate()
{
return date;
}

string Attendance::getTime()
{
return time;
}

Attendance::~Attendance()
{
cout<<"Attendance deleted!"<<endl;
}
```

Event.h

```
//contibutor - C.J Widanage - IT21244902

#include "Manger.h"

class Event{

    private:

      string eventid;

      int date;

      int time;

    public:

      Event();

      Event(string id, int d, int t);

      void setEventid(string id);

      void setDate(int d);

      void setTime(int t);

      char getEventid();

      int getDate();

      int getTime();

      ~Event();

};
```

Event.cpp

```
//contibutor - C.J Widanage - IT21244902

#include<iostream>

#include<string.h>
```

```
#include "Event.h"

using namespace std;


Event::Event(){

}

Event::Event(string id[], int d, int t){

   eventid = id;

   date=d;

   time=t;

}

void Admin::setEventid(stringid){

  eventid = id;

}

void Admin::setDate(int d){

  date=d;

}

void Admin::setTime(int t){

  time=t;

}

char Admin::getEventid(){

}

int Admin::getDate(){

}

int Admin::getTime(){

}
```

Feedback.h

```
#pragma once
//Contributor IT21288630 Kulathunga.R.G.G.P

#include <iostream>
#include <string>
using namespace std;

class Feedback
{
private:
	string fid;
	string status;

public:
	Feedback();
	Feedback(string id, string st);
	void setFid(string id);
	void setStatus(string st);
	string getFid();
	string getStatus();
	~Feedback();
};
```

Feedback.cpp

```
//Contributor IT21288630 Kulathunga.R.G.G.P

#include "Feedback.h"
#include <iostream>

Feedback::Feedback()
{
	fid = "";
	status = "";
}
```

```
Feedback::Feedback(string id, string st)
{
	fid = id;
	status = st;
}

void Feedback::setFid(string id)
{
	fid = id;
}

void Feedback::setStatus(string st)
{
	status = st;
}

string Feedback::getFid()
{
	return fid;
}

string Feedback::getStatus()
{
	return status;
}

Feedback::~Feedback()
{
	cout << "Feedback deleted" << endl;
}
```

Admin.h

```
#pragma once
//Contributor IT21288630 Kulathunga.R.G.G.P

#include "User.h"
#include "Feedback.h"
#define size 3



class Admin : public User
{
private:
	string adid;
	Feedback* fed[size];

public:
	Admin();
	Admin(string id, string uname, string pass, int num, string mail);
	void setAdid(string id);
	void addFeedback(Feedback* f);
	string getAdid();
	string viewFeedback();
	void addAccount();
	void removeAccount();
	~Admin();
};
```

Admin.cpp

```
//Contributor IT21288630 Kulathunga.R.G.G.P

#include "Admin.h"
```

```cpp
Admin::Admin()
{
	adid = "";
	username = "";
	password = "";
	phone = 0;
	email = "";
}

Admin::Admin(string id, string uname, string pass, int num, string mail)
{
	adid = id;
	username = uname;
	password = pass;
	phone = num;
	email = mail;
}

void Admin::setAdid(string id)
{
	adid = id;
}

void Admin::addFeedback(Feedback* f)
{

}

string Admin::getAdid()
{
	return adid;
}

Admin::~Admin()
{
	cout << "Admin deleted" << endl;
}
```

Main.cpp

```
//Contributor IT21288630 Kulathunga.R.G.G.P
//contributor - W.A.I.A.Bandara - IT21270710
//contibutor - C.J Widanage - IT21244902
//Contributor- H. M. N. D. S. Heenekenda -IT21323584
//contributor-Wijerathne.R.V.A.N.S -IT21264184

#include<iostream>
#include<string.h>
#include "Attendance.h"
#include "Manager.h"
#include "Resources.h"
#include "User.h"
#include "Feedback.h"
#include "Admin.h"
#include "Employee.h"
#include "Event.h"
using namespace std;


int main()
{
  User* us1 = new user("Kusun", "hack1234", "0774567833",
"Kusun@gmail.com");
    User* us1 = new user("Nimal", "sanka123", "0776783233",
"Nimal@gmail.com");
    User* us1 = new user("Shanaka", "gihan1234", "0759876342",
"Shanaka@gmail.com");


  Employee* em1 = new employee("e1", "1department");
  Employee* em1 = new employee("e2", "4department");
  Employee* em1 = new employee("e3", "3department");


  Attendance* at1 = new attendance("a1", "11/03/2020", "7.30AM");
  Attendance* at1 = new attendance("a1", "20/05/2020", "6.30PM");
  Attendance* at1 = new attendance("a1", "27/07/2020", "7.30AM");


  Feedback* fe1 = new Feedback("f1", "Good");
  Feedback* fe2 = new Feedback("f2", "Nice");
  Feedback* fe3 = new Feedback("f3", "Need improvement");
```

```
  Admin* ad1 = new Admin("a1", "Harry", "1234a", 0712354167,
"harry@gmail.com");
  Admin* ad2 = new Admin("a2", "Tobi", "4567b", 0712345671,
"tobi@gmail.com");


  Event* eve1 = new Event("ev1" , "12/04/2020" , "9.00AM") ;
  Event* eve2 = new Event("ev2" ,"23/05/2020"  , "10.30AM");
  Event* eve3 = new Event("ev3" , "20/06/2020" , "6.30PM" );


  Resources* res1= new Resources("re1","Recruiting Process
presentation slides");
  Resources* res1= new Resources("re2","Training and development
presentation slides");
  Resources* res1= new Resources("re3","Career path planing
presentation slides");


  Manager* mng1= new Manager("mang1","B.M.W.Rathanayake","Accounts
Department");
  Manager* mng1= new Manager("mang1","K.S.Karunarathne","Stores
Department");
  Manager* mng1= new
Manager("mang1","K.T.M.Basnayake","Correspondence Department");


  return 0;

}
```