



Topic : Online Teacher Trainer.

Group no : MLB_WE_PRORATA_06.

Campus : Malabe.

Submission Date: 18.11.2022

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
IT20126438	Colambage K.G	0752116335

Exercise 1

System Requirements

1. system users (The trainer (admin) and Training Teachers) can login to the system using their login credentials (username and password).
2. Only registered Training Teachers can login to the system with valid login credentials provided by the Training.
3. Admin can login to the system by providing pre-defined username and password. This username and password are only known by the admin.
4. Admin and teachers should have different login pages with different functions.
5. User can select their role (training or teachers) from the first page in order to navigate to the login page.
6. The admin should be able to **add, update, delete (remove), edit** teachers and teacher lists.
7. The admin can view a list of all the complete details of teacher present in the store.
8. Training teachers can view list of teachers available.
9. All the information added or removed will be updated in the database file (teacher.txt). Training Teachers can access and edit the database.

Noun & Verb Analysis

- Nouns in **BLUE** colour.

- Verbs in RED colour.

All the users can select their role (Trainer or Teacher) from the menu at the first page.

After selection of role, the user should login to the system by providing his/her username and password which allows them to access the the training management system.

The user can then select an option (services) from the menu that allocated to them according to their role.

If user logged in as trainer (admin) there will be different services such as,

- Add new teacher
- Update any teacher
- Delete teacher
- View list of all teachers

If trainer wants to add new teachers, he/she can select “Add new teacher” and provide teacher details such as teacher ID, teacher name, and author name.

Also, trainer can easily delete or update the teachers by selecting the option from the menu. If user logged in as a training teacher there will be a menu with different services such as,

- Borrow a teacher
- Return a teacher
- View List of teachers

User can select the option they want from the menu.

Identified Classes Using Noun Verb Analysis

Identified class

- User
- Trainer
- Teacher

Exercise 2

CRC Cards for Library Management System

User

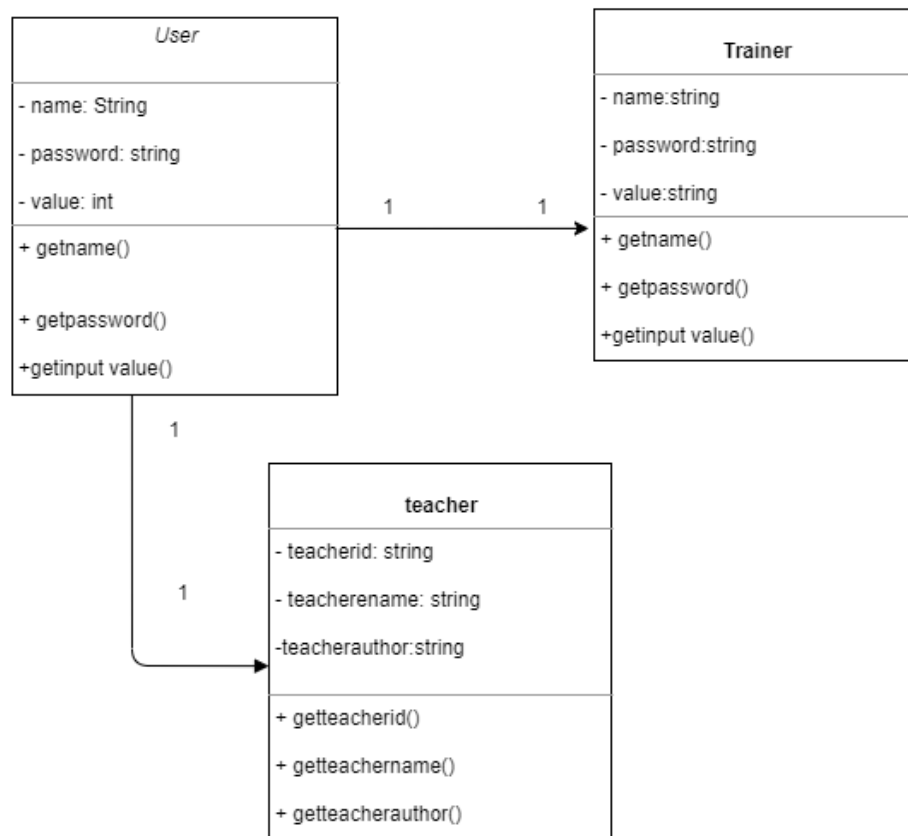
Responsibilities:	Collaborations:
Login to the system	
Borrow teacher	teacher
Return teacher	teacher

Trainer	
Responsibilities:	Collaborations:
Login to the system	
CRUD operation for Training details	teacher

Teacher	
Responsibilities:	Collaborations:
Store teacher details	

Exercise 3

Class Diagram



Exercise 4

Coding

Main.cpp

```
#include <iostream>
#include "class.h"
#include <cstring>
#include <fstream>
#include <sstream>
using namespace std;
```

```

int main (int argc, char **argv)
{

int omenuue = 0;

while (omenuue != 9)
{

system ("cls");

cout << "\n Online Teacher Trainer  \n\n";
cout << "\n1. Trainer";
cout << "\n2. User";
cout << "\n9. Exit";
cout << "\n\nEnter Your Value : ";
cin >> omenuue;

if (omenuue == 1)

{

char auser[10];
char apassword[10];

system ("cls");

cout << "\n ----- Trainer Login -----";
cout << "\n\n Please enter your username: ";
cin >> auser;
cout << "\n Please enter your
password:"; cin >> apassword;

if ((strcmp (auser, ADMINUSER) == 0) &&
(strcmp (apassword, ADMINPASSWORD) ==
0)) {

//getchar ();
//getchar ();

//system ("cls");

int imenuue = 0;

while (imenuue != 9)
{

cout <<
"\n\n Trainer Login Is Successful ! ";

```

```

cout << "\n1. Add New Teacher";
cout << "\n2. See Available Teachers";
cout << "\n3. Delete Teacher Record";
cout << "\n4. Update Teacher Details";
cout << "\n9. Go To Main Menu";
cout << "\n\n Enter Your Value : ";
cin >> imenue;

switch (imenue)
{

case 1:
addTeacher ();
break;

case 2:
listOfAllTeachers ();
break;

case 3:
deleteTeacher ();
break;

case 4:
updateTeacher ();
break;

default:
cout <<
"\n\n Invalid Value. Please Enter The Valid
Value"; getchar ();

}

system ("cls");

}
}
else
{
cout <<
"\n\n Invalid User Credentials. Please Correct User
Credentials\n\n Press any key to continue..";

getchar ();
getchar ();

}
}

else if (omenue == 2)

```

```

{

char uuser[10];

char upassword[10];


system ("cls");

cout << "\n ----- User Login -----";
cout << "\n\n Please enter your username: ";
cin >> uuser;
cout << "\n Please enter your password:";
cin >> upassword;


if ((strcmp (uuser, USER) == 0) &&
    (strcmp (upassword, USERPASSWORD) == 0))
{

int imenue = 0;

while (imenue != 9)
{

system ("cls");

cout << "\n1. See Available Teachers ";
cout << "\n2. Borrow a Teacher ";
cout << "\n3. Return a Teacher";
cout << "\n9. Go To Main Menu";
cout << "\n\n Enter your choice : ";
cin >> imenue;


switch (imenue)
{

case 1:
listOfAllTeachers ();
Break;

case 2:
borrowTeacher ();
break;

case 3:
returnTeacher ();
break;

case 9:
break;

```



```

default:
cout <<
"\n\n Error : Invalid Value. Please Enter The Valid Value";

getchar ();
getchar ();

}

}

system ("cls");

} else {

cout <<
"\n\n Invalid User Credentials. Please Correct User
Credentials\n\n Press any key to continue..";

getchar ();
getchar ();

}
}

else if (omenu != 9)
{

cout << "\n\n Invalid Value. Press any key to continue..";

getchar ();
getchar ();

} else {

cout << "\n\n Thank you for using it !! Press any key to exit";

getchar ();
getchar ();

}

}

return 0;
}

int addTeacher ()
{

```

```

char teacherid[4];
char teachername[20];
char teacherauthor[20];
int year, month, day;

time_t t = time (NULL);
struct tm tm = *localtime (&t);
year = tm.tm_year + 1900;
month = tm.tm_mon + 1;
day = tm.tm_mday;

system ("cls");
cout << "\n ---- Add a new Teacher ----";
cout << "\n\n Please Enter 4 Digit teacher
ID :"; cin >> teacherid;
cout << "\n Please Enter teacher Name
: "; cin >> teachername;
cout << "\n Please Enter Teacher Author
: "; cin >> teacherauthor;

ofstream file;
file.open("teacher.txt", ios::app);

file << teacherid << " ";
file << teachername << " ";
file << teacherauthor << " ";
file << year << " " ;
file << month << " " ;
file << day << "\n";
file.close ();
cout << "\n Teacher Added Successfully
!"; cout << "\n Press Eny Key To
continue.."; getchar ();
getchar ();

}

int listOfAllTeachers ()
{
cout << '\n';
string line;
ifstream myfile ("teacher.txt");
if (myfile.is_open())
{
while ( getline (myfile,line) )
{
cout << line << '\n';
}
}
myfile.close();
cout << "\n Press Eny Key To

```

```

continue.."; getchar ();
getchar ();
}
}

int deleteTeacher ()
{ }

int updateTeacher ()
{ }

int borrowTeacher ()
{
}

int returnTeacher ()
{
}

int adminLogin ()
{
}

int userLogin ()
{
}

```

Class.h

```

#define ADMINUSER "Trainer" #define
ADMINPASSWORD "trainer123"

#define USER "kavindu"
#define USERPASSWORD "kavindu123"

#define FINE 500

int addTeacher();

int deleteteacher();

int updateteacher();

int listOfAllTeachers();

int borrowteacher();

int returnTeacher();

int adminLogin();

int userLogin();

```