

**Konstantinos Kousounnis
Java Application PrivateSchoolSA
Documentation**

Project description:

Kousounnis Konstantinos

Implementation of a private school structure which will consists of Courses, Students, Trainers, Assignments.

Project requirements

- 1) The application must ask from the command prompt to input data to all the entities and it should give the option to add more than one entry at a time.
- 2) If the user decides not to type any data, the program should use synthetic data.
- 3) A list of all the students.
- 4) A list of all the trainers.
- 5) A list of all the assignments.
- 6) A list of all the courses.
- 7) All the students per course.
- 8) All the trainers per course.
- 9) All the assignments per course.
- 10) All the assignments per student.
- 11) A list of students that belong to more than one courses.

12) Lastly, the program should ask from the user a date and it should output a list of students who need to submit one or more assignments on the same calendar week as the given date.

- Explain: The calendar week is considered as Monday to Friday for submissions. For example, if the user inputs 15/2/2019 you need to check from Monday 11/2/2019 to Friday 15/2/2019. If the user inputs 16/2/2019 (Saturday) or 17/2/2019 (Sunday) you need to check from Monday 11/2/2019 to Friday 15/2/2019.

Design Structure

Design Structure PrivateSchoolSA

Design of a Private School Structure

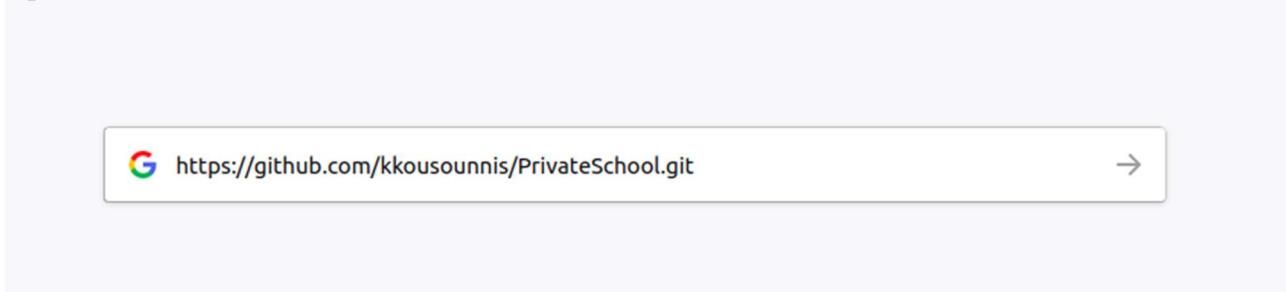
Course	Trainer
<p>-title -stream -type -start_date -end_date</p>	<p>-firstName -lastName -subject</p>
<p>+getTitle +setTitle +getStream +setStream +getType +setType +getStartDate +setStartDate +getEndDate +setEndDate</p>	<p>+getFirstName +setFirstName +getLastName +setLastName</p>

Student	Assignment
<p>-firstName -lastName -dateOfBirth -tuitionFees</p>	<p>-title -description -subDateTime -oralMark -totalMark</p>
<p>+getFirstName +setFirstName +getLastname +setLastName +getDateOfBirth + setDateOfBirth +getFees +setFees</p>	<p>+getTitle +setTitle +getDescription +setDescription +getSubDateTime +setSubDateTime +getOralMark +setOralMark +getTotalMark +setTotalMark</p>

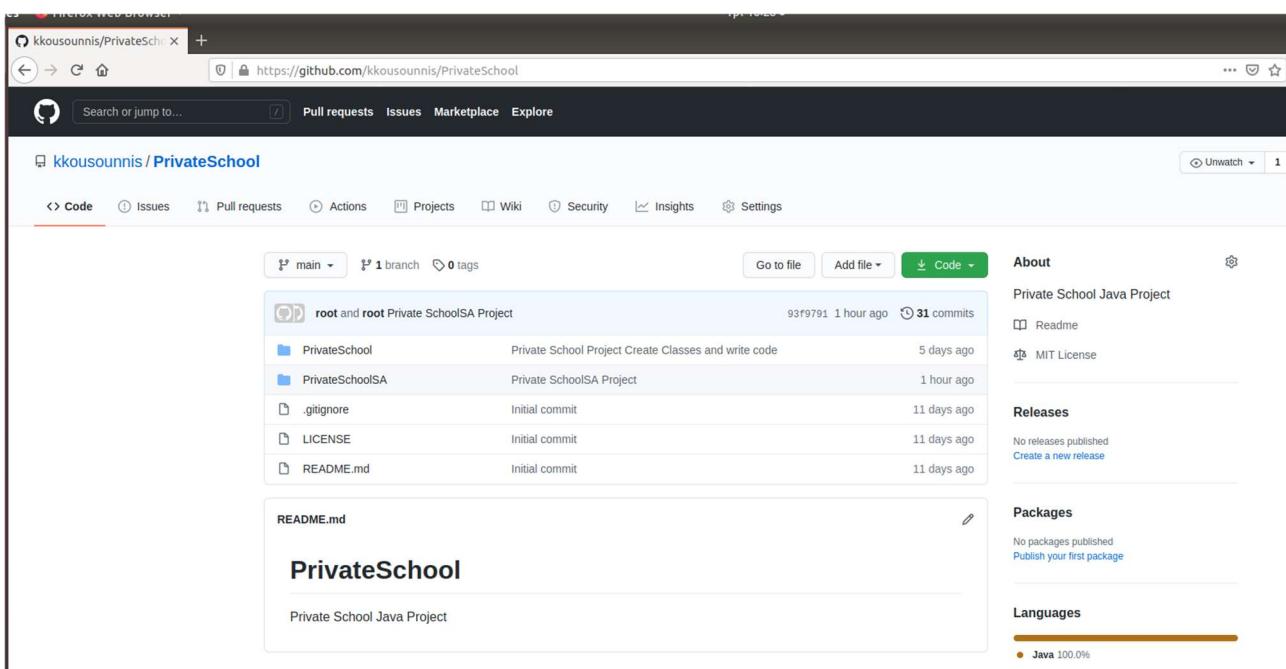
Documentation Install Project Private SchoolSA

Github

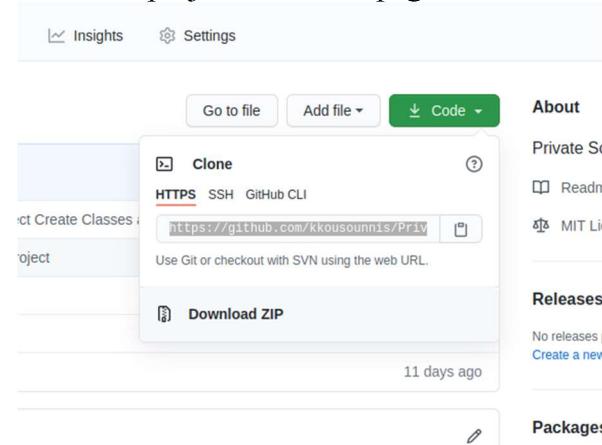
Open a search browser and search with url below.



<https://github.com/kkousounnis/PrivateSchool>



Open git command line or download project from web page.



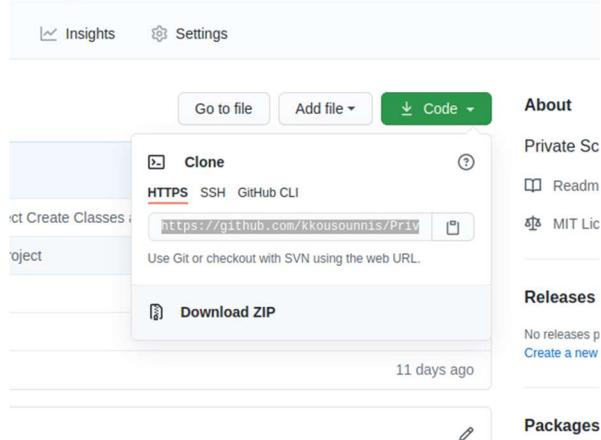
1) Command line
Git clone <https://github.com/kkousounnis/PrivateSchool.git>

type

```
File Edit View Search Terminal Help  
konstantinos@konstantinos-Alienware-15-R2:~$ git clone https://github.com/kkousounnis/PrivateSchool.git
```

2)

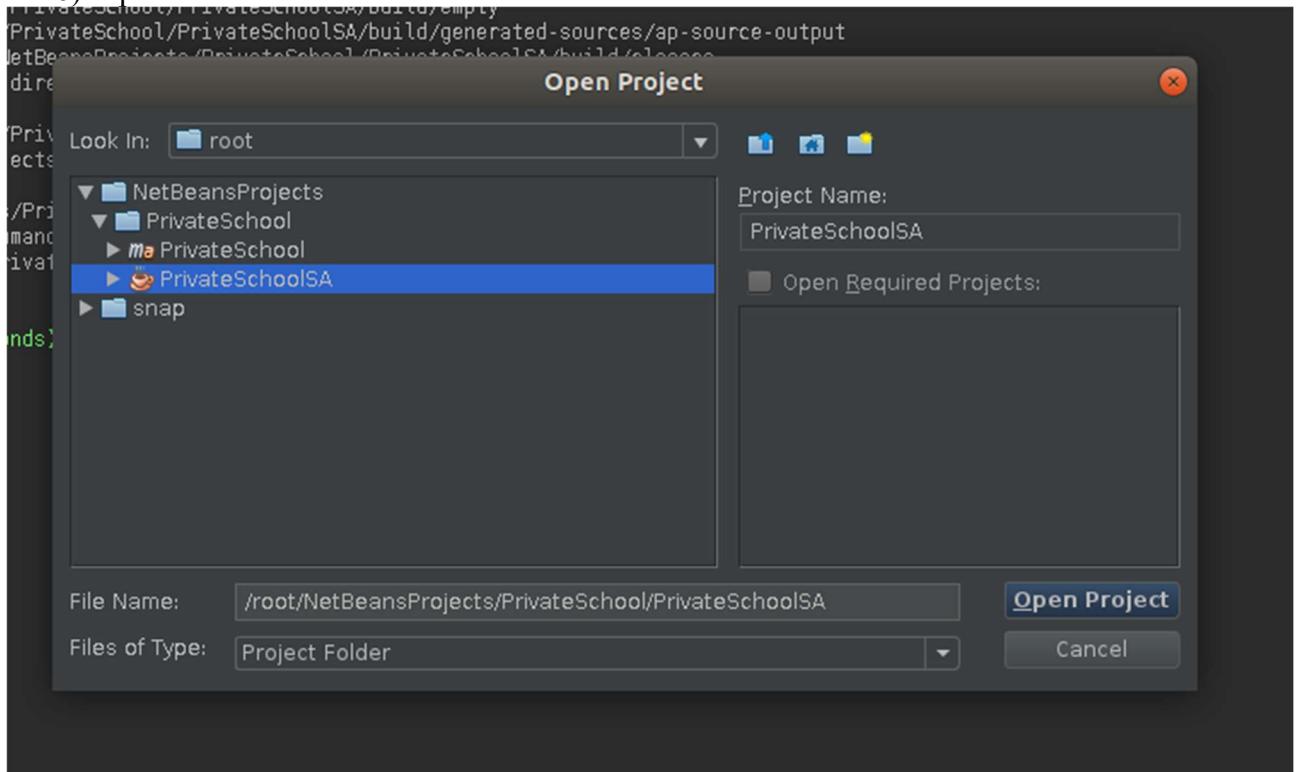
3) Download project from github



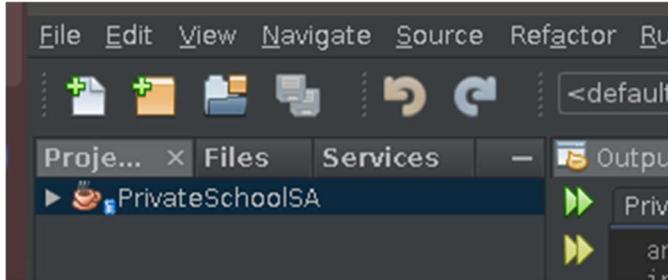
4) Download Zip

5) Extract project

6) open Netbeans



We want to open PrivateSchoolSA project the other one is an older version of the completed project.

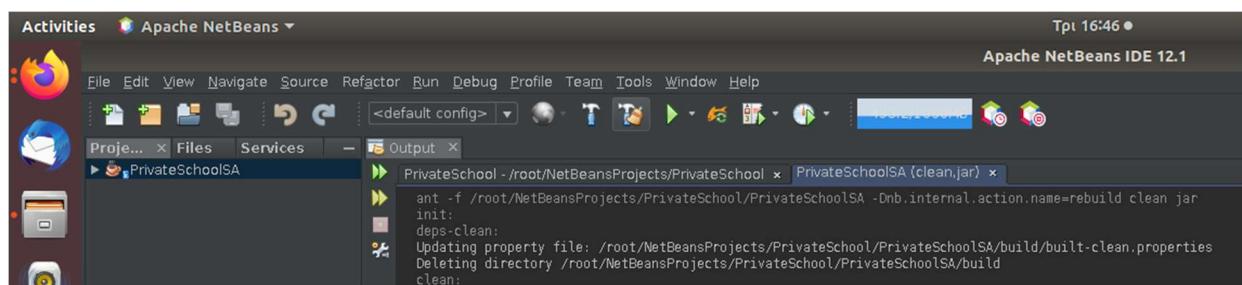


When we are all set we shall see at projects folder **PrivateSchoolSA** project.

Running Program

First build project.

1. shift + alt +f11 one way
2. Clean Build hummer icon choose.

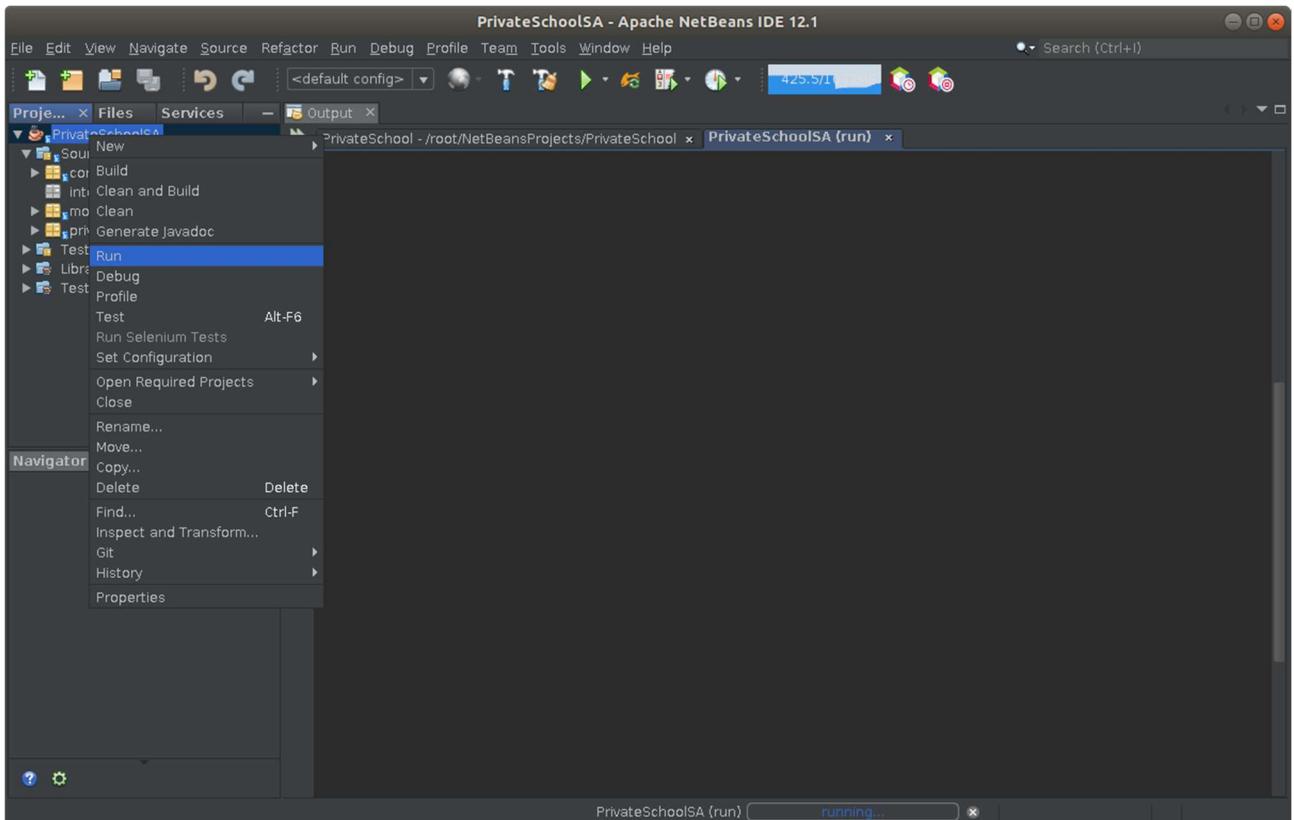
A screenshot of the NetBeans IDE interface showing the 'Output' window. The log entries show a successful build:

```
ant -f /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA -Dnb.internal.action.name=rebuild clean jar
init:
deps-clean:
Updating property file: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build/built-clean.properties
Deleting directory /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build
clean:
init:
deps-jar:
Created dir: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build
Updating property file: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build/built-jar.properties
Created dir: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build/classes
Created dir: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build/empty
Created dir: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build/generated-sources/ap-source-output
Compiling 17 source files to /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build/classes
Copied 1 empty directory to 1 empty directory under /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build
compile:
Created dir: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/dist
Copying 1 file to /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/build
Nothing to copy.
Building jar: /root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/dist/PrivateSchoolSA.jar
To run this application from the command line without Ant, try:
java -jar "/root/NetBeansProjects/PrivateSchool/PrivateSchoolSA/dist/PrivateSchoolSA.jar"
deploy:
jar:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Build succesful.

Now run PrivateSchoolSA

1. Press f6
2. Select project → right click → run.



The project now has started and we will our Menu at the command line.

```
---Welcome to Private SchoolsSA where every dream becomes a reality---
---
---
Description of School
This is a private school which offers intensive lessons for someone who desires to learn.
Please choose how would you like to proceed.
1: Proceed to Main Menu.
2: Create fake data and proceed to Main Menu.
3: Terminate.
Enter a number to proceed :
```

Project Requirements

- The application must ask from the command prompt to input data to all the entities and it should give the option to add more than one entry at a time.
- If the user decides not to type any data, the program should use synthetic data.

We have three choices to meet our requirements.

1st choice continue without any synthetic data the user will have to create all fields manually.

2nd choice continue with synthetic data when selected the java program will create fake data and then user can create more or can use the ones that have been already made.

3rd choice terminate our program exit application.

Continue First Choice

If we type something else we will be asked to retry.

```
--Welcome to Private SchoolsSA where every dream becomes a reality---  
---  
---  
  
Description of School  
This is a private school which offers intensive lessons for someone who desires to learn.  
Please choose how would you like to proceed.  
1: Proceed to Main Menu.  
2: Create fake data and proceed to Main Menu.  
3: Terminate.  
Enter a number to proceed : 9
```

Type 9

```
1: Proceed to Main Menu.  
2: Create fake data and proceed to Main Menu.  
3: Terminate.  
Enter a number to proceed : 9  
Enter a valid Integer value  
Enter a number to proceed :
```

A message will appear to retry as well as if we type sentences instead of a number.

```
2: Create fake data and proceed to Main Menu  
3: Terminate.  
Enter a number to proceed : 9  
Enter a valid Integer value  
Enter a number to proceed : gdrdere  
Enter a valid Integer value  
Enter a number to proceed :
```

Press 1 and continue.

```
Main Menu  
Please choose how would you like to proceed  
1: Proceed to Course Menu.  
2: Proceed to Student Menu.  
3: Proceed to Trainer Menu.  
4: Proceed to Assignment Menu.  
5: Return to strat Menu.  
6: Terminate.  
Enter a number to proceed : |
```

Another Menu appears and gives us 6 choices with the same way we move through our program
Select 1 Course Menu there are security checks to every input.

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : |
```

Another menu appears with 6 choices

Because I have no courses yet first I will try to see if I have any courses by typing 2 and I will see a table with no courses.

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : 2

---Courses---
Course Menu
Please choose how would you like to proceed
1: Course Manually create
```

The table ---Courses--- is empty no courses yet

Create a course select 1.

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : 1
Choose the way you want to insert your values
Type 1 for step by step input Press 2 for multiple inputs
Enter a number to proceed : |
```

Requirement 2) The application must ask from the command prompt to input data to all the entities and it should give the option to add more than one entry at a time.

We can choose how the user inputs the data

1st select 1 step by step

```
Please give the title of course.  
course1  
Please give the name of course  
Javascript  
Please type (f) if it is FullTime or type (p) if it is PartTime  
f  
Please give me the start date of the course  
please enter date like this (DD/MM/YYYY)  
01/01/2021  
Please give me the ending date of the course  
please enter date like this (DD/MM/YYYY)  
01/02/2021  
Course Menu  
Please choose how would you like to proceed  
1: Course Manually create.  
2: Show all courses.  
3: Show students per course.  
4: Show trainers per course.  
5: Show assignments per course.  
6: Return.  
Enter a number to proceed : |
```

User types step by step by asking him every time for every field
My example is as in screen

1. course1
2. Javascript
3. f (Stands for fulltime)
4. 01/01/2021
5. 01/02/2021

For every field there is a security check.

I can't write multiple words to any fields.

```
Please give the title of course.  
course1 course2 course3  
There must be only one word, spaces aren't allowed.Try again.  
course1|
```

Error message appear please retype.

When I select (f) or (p) only those two values are acceptable nothing else as we can see bellow.

```
Please give the title of course.  
course1 course2 course3  
There must be only one word, spaces aren't allowed.Try again.  
course1  
Please give the name of course  
Go mpla mpla  
There must be only one word, spaces aren't allowed.Try again.  
Go  
Please type (f) if it is FullTime or type (p) if it is PartTime  
t80  
Please type (f) if it is FullTime or type (p) if it is PartTime  
5  
Please type (f) if it is FullTime or type (p) if it is PartTime  
q  
Please type (f) if it is FullTime or type (p) if it is PartTime  
p  
Please give me the start date of the course  
please enter date like this (DD/MM/YYYY)  
|
```

Security Check also occurs in date we must type dd/mm/yyyy nothing else will the program accept.

```
Please type (f) if it is FullTime or type (p) if it is PartTime  
p  
Please give me the start date of the course  
please enter date like this (DD/MM/YYYY)  
dhsidhhjsdsh  
please enter date like this (DD/MM/YYYY)  
1221/212121  
please enter date like this (DD/MM/YYYY)  
01/01/2021  
Please give me the ending date of the course  
please enter date like this (DD/MM/YYYY)
```

Error message until right input.

```
p  
Please give me the start date of the course  
please enter date like this (DD/MM/YYYY)  
dhsidhhjsdsh  
please enter date like this (DD/MM/YYYY)  
1221/212121  
please enter date like this (DD/MM/YYYY)  
01/01/2021  
Please give me the ending date of the course  
please enter date like this (DD/MM/YYYY)  
01/02/2021  
Course Menu  
Please choose how would you like to proceed  
1: Course Manually create.  
2: Show all courses.  
3: Show students per course.  
4: Show trainers per course.  
5: Show assignments per course.  
6: Return.  
Enter a number to proceed : |
```

We have returned to Menu after we inserted Course and now select 2 to see course.

I have repeated two times procedure since I wanted to show the security checks.

```

---Courses---
1: Course{ title=coursel , stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=01-February-2021}
2: Course{ title=coursel, stream=Go, type=PartTime, startDate=01-January-2021, endDate=01-February-2021}
    Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : |

```

Now we have two courses.

2nd Create courses with multiple choice.

So I will press 1 and the 2 for multiple input.

```

[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type {f} if it is FullTime or type {p} if it is PartTime

```

At first line I see how the input must be made.

```

[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type {f} if it is FullTime or type {p} if it is PartTime
course2 Postgresql f 01/01/2021 01/03/2021

```

So the way it must be typed is

course2 Postgresql f 01/01/2021 01/03/2021

Security check occer here as well the form is like this we aren't allowed any other way the words must be five and if fulltime or parttime or date is wrong we will be asked to retype afterwards.

```

[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type {f} if it is FullTime or type {p} if it is PartTime
course1 Mongodb Postgresql f p 01/01/2021 02/02/2021 djdsj
[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type {f} if it is FullTime or type {p} if it is PartTime
|
```

If we type more inputs or less wrong message appear to retype.

```
[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type (f) if it is FullTime or type (p) if it is PartTime
course1 Mongodb Postgresql f p 01/01/2021 02/02/2021 djdsj
[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type (f) if it is FullTime or type (p) if it is PartTime
course2 Mongodb v 01/djj 01/dieojd
Please type (f) if it is FullTime or type (p) if it is PartTime
```

If we type instead of f or p error message retype.

```
[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type (f) if it is FullTime or type (p) if it is PartTime
course1 Mongodb Postgresql f p 01/01/2021 02/02/2021 djdsj
[coursetitle stream {f}/{p} startdate enddate]
Warning there must be absolutely five inputs if not the system will ask you to retype all five inputs
Warning for third input.Please type (f) if it is FullTime or type (p) if it is PartTime
course2 Mongodb v 01/djj 01/dieojd
Please type (f) if it is FullTime or type (p) if it is PartTime
p
please enter date like this (DD/MM/YYYY)
```

Also date retype if it is not correct. I typed **01/djj** and asks me for right format.

The same check occur in all inputs courses, students, trainers, assignments

Only in assignments I allow to input a whole sentence in a field because we we need assignment description.

In the assignment section as explained 15 words are allowed for description section

```
Choose the way you want to insert your values
Type 1 for step by step input Press 2 for multiple inputs
Enter a number to proceed : 2

Type like this ->[assignmenttitle assignmentdate description(sentence) ]

Warning there must be three inputs but in third input we can have a sentence consisted of 12 words.
So you are allowed to type maximum 15 words.
If they are more than fifteen words the system will ask you to retype all three inputs consisted of fifteen words.
For exampel type: --> Assignment1 01/01/2021 Create a simple java program that prints hello world.
```

```
Warning there must be three inputs but in third input we can have a sentence consisted of 12 words.
So you are allowed to type maximum 15 words.
If they are more than fifteen words the system will ask you to retype all three inputs consisted of fifteen words.
For exampel type: --> Assignment1 01/01/2021 Create a simple java program that prints hello world.
```

Also afterward I have select to which course it belongs by typing number.

```
---Courses---
1: Course{ title=course1 , stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=01-February-2021}
2: Course{ title=course1, stream=Go, type=PartTime, startDate=01-January-2021, endDate=01-February-2021}
3: Course{ title=course2, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=01-February-2021}
4: Course{ title=course2, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=01-March-2021}
5: Course{ title=course2, stream=Mongodb, type=PartTime, startDate=01-February-2021, endDate=01-March-2021}
Please tell to which course assignment belongs
1|
```

The same procedure occurs for trainers and students.

Print course List.

```
Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : 2
```

```
---Courses---
1: Course{ title=course1 , stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=01-February-2021}
2: Course{ title=course1 , stream=Go, type=PartTime, startDate=01-January-2021, endDate=01-February-2021}
3: Course{ title=course2, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=01-February-2021}
4: Course{ title=course2, stream=Postgresql, type=FullTime, startDate=01-January-2021, endDate=01-March-2021}
5: Course{ title=course2, stream=Mongodb, type=PartTime, startDate=01-February-2021, endDate=01-March-2021}

    Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : |
```

The courses are now printed in a list here we have just made 5 courses and we can see all required fields.

The same procedure to show and create courses, students, trainers, and assignments can execute with the same way.

Example

```
Choose the way you want to insert your values
Type 1 for step by step input Press 2 for multiple inputs
Enter a number to proceed : 1
Please give me students first name.
konstantinos
Please give me students last name.
antreou
Please give me students date of birth.Type date of birth like this (DD/MM/YYYY).
please enter date like this (DD/MM/YYYY)
12/02/1995
Please give me the tuition fees of student.
4000|
```

For
last
field

tuition fees security also happens. If we type integer bigger than zero.

The same procedure is for every other with their security checks.

Show students who have more than one course example without having any course.

```
---Courses---
We must have more than two courses assigned.

Please go to course menu and assign more than 1 course.Then go to student menu and assign students to more than one course
Student Menu
Please choose how would you like to proceed
1: Create Manually student.
2: Show all students.
3: Choose student which lessons will attend.
4: Show student assignments
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed :
```

Privat

As we can see a message appear that we need more than two courses to see students.

The same messages appear for the other requirements that if we have no course we can't execute the task.

Example

Assign assignment to course.

```
---No courses have been assigned yet.---

Trainer Menu
Please choose how would you like to proceed.
1: Create Manually trainer.
2: Show all trainers
3: Choose trainer which course teaches.
4: Return.
Enter a number to proceed : |
```

As we see no courses have been assigned yet.

Synthetic Data 2nd Choice

We can create manually everything as we have seen above and see all project requirements. However we will create synthetic data so as to see that we are able to execute all project requirements.

Type 2 and fake data will be created.

```
---Welcome to Private SchoolsSA where every dream becomes a reality---
---
---
Description of School
This is a private school which offers intensive lessons for someone who desires to learn.
Please choose how would you like to proceed.
1: Proceed to Main Menu.
2: Create fake data and proceed to Main Menu.
3: Terminate.
Enter a number to proceed : 2
```

A list of all the students.

```
--Students--  
1: Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}  
2: Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}  
3: Student{TabinaSteinhammer, dateOfBirth=20-August-1992, tuitionFees=5000}  
4: Student{JohannaStanislaw, dateOfBirth=27-March-1994, tuitionFees=5000}  
5: Student{LeilaPierce, dateOfBirth=04-October-1993, tuitionFees=5000}  
6: Student{MartenKowalski, dateOfBirth=05-September-1995, tuitionFees=5000}  
7: Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}  
8: Student{SandersonTankard, dateOfBirth=07-April-1992, tuitionFees=5000}  
9: Student{LindsayMatches, dateOfBirth=10-February-1995, tuitionFees=5000}  
10: Student{TeadorMiddup, dateOfBirth=27-November-1995, tuitionFees=5000}  
  
Student Menu  
Please choose how would you like to proceed  
1: Create Manually student.  
2: Show all students.  
3: Choose student which lessons will attend.  
4: Show student assignments  
5: Show students who attend in more than one course.  
6: Return.  
Enter a number to proceed : |
```

A list of all the trainers.

```
Trainers  
1: Trainer{CorissaAbrahim, subject=Backend}  
2: Trainer{JudithHowatt, subject=DataBase}  
3: Trainer{MerryKenewell, subject=Backend}  
4: Trainer{IsahellaGockelen, subject=DataBase}  
5: Trainer{JudithHowatt, subject=Backend}  
6: Trainer{MerryKenewell, subject=DataBase}  
7: Trainer{DenysLambden, subject=Backend}  
8: Trainer{NealHairsnape, subject=DataBase}  
  
Trainer Menu  
Please choos how would you like to proceed.  
1: Create Manually trainer.  
2: Show all trainers  
3: Choose trainer which course teaches.  
4: Return.  
Enter a number to proceed :
```

A list of all the assignments.

```
Assignments  
1: Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}  
2: Assignment{title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, oralMark=0, totalMark=0}  
3: Assignment{title=Assignment3, description=Create an Assignment3, subDateTime=24-January-2021, oralMark=0, totalMark=0}  
4: Assignment{title=Assignment4, description=Create an Assignment4, subDateTime=02-February-2021, oralMark=0, totalMark=0}  
  
Assignment Menu  
Please choose how would you like to proceed  
1: Create manually assignment.  
2: Show all assignments.  
3: Choose course for assignment.  
4: Give date and show students with all relevant student assignments.  
5: Return.  
Enter a number to proceed :
```

Priv

A list of all the courses.

```

---Courses---
1: Course{ title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
2: Course{ title=Course2, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-March-2021}
3: Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
4: Course{ title=Course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
5: Course{ title=Course1, stream=PostgreSql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
6: Course{ title=Course1, stream=PostgreSql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
7: Course{ title=Course1, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
8: Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}

    Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed :

```

All the students per course.

```

... Students Per Course ...
Course{ title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
[Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}, Student{RinavAssilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}]
Course{ title=Course2, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-March-2021}
[Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}, Student{TabeinSteinhammer, dateOfBirth=20-August-1992, tuitionFees=5000}]
Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
[Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}, Student{RinavAssilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}, Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}]
Course{ title=Course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}, Student{RinavAssilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}, Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}, Student{LindsayMatches, dateOfBirth=10-February-1996, tuitionFees=5000}]
Course{ title=Course1, stream=PostgreSql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
[Student{Johannastanisla, dateOfBirth=27-March-1994, tuitionFees=5000}, Student{IsahellaGockelen, dateOfBirth=07-April-1992, tuitionFees=5000}]
Course{ title=Course1, stream=PostgreSql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Student{Johannastanisla, dateOfBirth=27-March-1994, tuitionFees=5000}, Student{IsahellaGockelen, dateOfBirth=07-April-1992, tuitionFees=5000}, Student{LindsayMatches, dateOfBirth=10-February-1996, tuitionFees=5000}]
Course{ title=Course1, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
[Student{MartenKowalski, dateOfBirth=05-September-1995, tuitionFees=5000}, Student{TeodorMidup, dateOfBirth=27-November-1995, tuitionFees=5000}]
Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Student{MartenKowalski, dateOfBirth=05-September-1995, tuitionFees=5000}, Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}]

    Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed :

```

All the trainers per course.

```

---Trainer Per Course---
Course{ title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}
[Trainer{MerryKenewell, subject=Backend}]

Course{ title=Course2, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-March-2021}
[Trainer{CorissaAbrahim, subject=Backend}]

Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
[Trainer{DenysLambden, subject=Backend}]

Course{ title=Course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Trainer{JudithHowatt, subject=Backend}]

Course{ title=Course1, stream=PostgreSql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
[Trainer{NealHairsnape, subject=DataBase}]

Course{ title=Course1, stream=PostgreSql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Trainer{MerryKenewell, subject=DataBase}]

Course{ title=Course1, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
[Trainer{IsahellaGockelen, subject=DataBase}]

Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Trainer{JudithHowatt, subject=DataBase}]

    Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed :

```

All the assignments per course.

```

...Assignments Per Course...
Course[title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course2, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-March-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course2, stream=Go, type=Fulltime, startDate=01-January-2021, endDate=31-May-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course1, stream=PostgreSQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course1, stream=PostgreSQL, type=PartTime, startDate=01-January-2021, endDate=31-May-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course1, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora
Course[title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021]
[Assignment[title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0], Assignment[title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, ora

        Course Menu
Please choose how would you like to proceed
1: Course Manually create.
2: Show all courses.
3: Show students per course.
4: Show trainers per course.
5: Show assignments per course.
6: Return.
Enter a number to proceed : |
```

Here in order to see all assignments we must shift right.

All the assignments per student.

```

Student Menu
Please choose how would you like to proceed
1: Create Manually student.
2: Show all students.
3: Choose student which lessons will attend.
4: Show student assignments
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed : 4
```

```

---Students---
1: Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}
2: Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}
3: Student{TabinaSteinhammer, dateOfBirth=20-August-1992, tuitionFees=5000}
4: Student{JohannaStanislaw, dateOfBirth=27-March-1994, tuitionFees=5000}
5: Student{LeilaPierce, dateOfBirth=04-October-1993, tuitionFees=5000}
6: Student{MartenKovalski, dateOfBirth=05-September-1995, tuitionFees=5000}
7: Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}
8: Student{SandersonTankard, dateOfBirth=07-April-1992, tuitionFees=5000}
9: Student{LindsayMatches, dateOfBirth=10-February-1995, tuitionFees=5000}
10: Student{TeadorMiddup, dateOfBirth=27-November-1995, tuitionFees=5000}
Please specify student from List in order to see his assignments by typing number
Enter a number to proceed :
```

First select student type number.

```

9: Student{LindsayMatches, dateOfBirth=10-Feb
10: Student{TeadorMiddup, dateOfBirth=27-Nov
Please specify student from List in order to proceed
Enter a number to proceed : 5

```

```

---Students Per Assignment---
Student{LeilaPierce, dateOfBirth=04-October-1993, tuitionFees=5000}
Course{ title=Course1, stream=PostgreSQL, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
[Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0}, Assignment{title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, oralMark=0}]

Student Menu
Please choose how would you like to proceed
1: Create Manually student.
2: Show all students.
3: Choose student which lessons will attend.
4: Show student assignments
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed :

```

We have based the student we gave his assignments.
A list of students that belong to more than one courses.

```

Student Menu
Please choose how would you like to proceed
1: Create Manually student.
2: Show all students.
3: Choose student which lessons will attend.
4: Show student assignments
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed : 5

```

Show students who attend multiple courses.

```

--Students who Attend Multiple Courses---
Student{KlemensBobak, dateOfBirth=05-December-1994, tuitionFees=5000}
Attends the courses below:
[Course{ title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}, Course{ title=Course2, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-March-2021}, Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course2, stream=PostgreSQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course2, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}]

Student{AnnaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}
Attends the courses below:
[Course{ title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}, Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course2, stream=PostgreSQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course2, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}]

Student{JohannaStanislaw, dateOfBirth=27-March-1994, tuitionFees=5000}
Attends the courses below:
[Course{ title=Course1, stream=PostgreSQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}]

Student{MarekKowalski, dateOfBirth=05-September-1995, tuitionFees=5000}
Attends the courses below:
[Course{ title=Course1, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}]

Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}
Attends the courses below:
[Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}, Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}]

Student Menu
Please choose how would you like to proceed
1: Create Manually student.
2: Show all students.
3: Choose student which lessons will attend.
4: Show student assignments
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed :

```

We can see based what courses students have taken if they have multiple courses we see their names.
We can set every time we create a student or trainer or assignment to which course it belongs and also we can select to choose from list for multiple.

Example

```

Student Menu
Please choose how would you like to proceed
1: Create Manually student.
2: Show all students.
3: Choose student which lessons will attend.
4: Show student assignments
5: Show students who attend in more than one course.
6: Return.
Enter a number to proceed : 3

```

3: Choose student which lesson will attend. We can select multiple courses.

```
--Students--  
1: Student{KlemensHabgood, dateOfBirth=05-December-1994, tuitionFees=5000}  
2: Student{RinaVasilchenko, dateOfBirth=06-June-1994, tuitionFees=5000}  
3: Student{TabinaSteinhammer, dateOfBirth=20-August-1992, tuitionFees=5000}  
4: Student{JohannaStanislaw, dateOfBirth=27-March-1994, tuitionFees=5000}  
5: Student{LeilaPierce, dateOfBirth=04-October-1993, tuitionFees=5000}  
6: Student{MartenKowalski, dateOfBirth=05-September-1995, tuitionFees=5000}  
7: Student{ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000}  
8: Student{SandersonTankard, dateOfBirth=07-April-1992, tuitionFees=5000}  
9: Student{LindsayMatches, dateOfBirth=10-February-1995, tuitionFees=5000}  
10: Student{TeadorMiddup, dateOfBirth=27-November-1995, tuitionFees=5000}  
Please specify student from List by typing number  
Enter a number to proceed : |
```

First we choose which students we want by typing number. Example 3

```
--Courses--  
1: Course{ title=Course2, stream=Javascript, type=FullTime, startDate=01-January-2021, endDate=31-March-2021}  
2: Course{ title=Course2, stream=Javascript, type=PartTime, startDate=01-January-2021, endDate=31-March-2021}  
3: Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}  
4: Course{ title=Course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}  
5: Course{ title=Course1, stream=PostgreSql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}  
6: Course{ title=Course1, stream=PostgreSql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}  
7: Course{ title=Course1, stream=Mysql, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}  
8: Course{ title=Course1, stream=Mysql, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}  
Please tell me to which course will the student attend please type number of course.  
Enter a number to proceed : 7
```

Then we see all courses and we choose which course attends. Example 7

The same procedure for every other requirement assign trainer to course, assignment to course.

Lastly, the program should ask from the user a date and it should output a list of students who need to submit one or more assignments on the same calendar week as the given date.

Select 4 Proceed to Assignment Menu

```
Main Menu  
Please choose how would you like to proceed  
1: Proceed to Course Menu.  
2: Proceed to Student Menu.  
3: Proceed to Trainer Menu.  
4: Proceed to Assignment Menu.  
5: Return to strat Menu.  
6: Terminate.  
Enter a number to proceed : 4|
```

Select 2 to see all dates of assignments locate a date and then typed show we can see some students. If you type an irrelevant date no students will be printed.

```
Assignments
1: Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}
2: Assignment{title=Assignment2, description=Create an Assignment2, subDateTime=16-January-2021, oralMark=0, totalMark=0}
3: Assignment{title=Assignment3, description=Create an Assignment3, subDateTime=24-January-2021, oralMark=0, totalMark=0}
4: Assignment{title=Assignment4, description=Create an Assignment4, subDateTime=02-February-2021, oralMark=0, totalMark=0}
    Assignment Menu
Please choose how would you like to proceed
1: Create manually assignment.
2: Show all assignments.
3: Choose course for assignment.
4: Give date and show students with all relevant student assignments.
5: Return.
Enter a number to proceed : 2
```

Select 4 for the user to give a date and output a list of students who need to submit one or more assignments on the same calendar week as the given date.

```
Assignment Menu
Please choose how would you like to proceed
1: Create manually assignment.
2: Show all assignments.
3: Choose course for assignment.
4: Give date and show students with all relevant student assignments.
5: Return.
Enter a number to proceed : 4
```

```
Please enter the date you wish to see all relevant student assignments
Date must be typed like this (DD/MM/YYYY)
```

We are asked to give a relevant date.

First lets type an irrelevant date

```
Please enter the date you wish to see all relevant student assignments
Date must be typed like this (DD/MM/YYYY)
15/11/2021
```

15/11/2021 irrelevant date correctly nothing is printed and we have returned to Assignment Menu

```
Please enter the date you wish to see all relevant student assignments
Date must be typed like this (DD/MM/YYYY)
15/11/2021
    Assignment Menu
Please choose how would you like to proceed
1: Create manually assignment.
2: Show all assignments.
3: Choose course for assignment.
4: Give date and show students with all relevant student assignments.
5: Return.
Enter a number to proceed : |
```

Type relevant date 07/01/2021

```

Output [PrivateSchool - /root/NetBeansProjects/PrivateSchool x | PrivateSchoolSA (run) x]
Johanna Stanislaw has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[LeilaPierce, dateOfBirth=04-October-1993, tuitionFees=5000]
Course{ title=Course1, stream=PostgreSQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Leila Pierce has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[MartenKowalski, dateOfBirth=05-September-1995, tuitionFees=5000]
Course{ title=Course1, stream=MySQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Marten Kowalski has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[MartenKowalski, dateOfBirth=05-September-1995, tuitionFees=5000]
Course{ title=Course1, stream=MySQL, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Marten Kowalski has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000]
Course{ title=Course2, stream=Go, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Elladine Shorrock has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[ElladineShorrock, dateOfBirth=22-August-1996, tuitionFees=5000]
Course{ title=Course1, stream=MySQL, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Elladine Shorrock has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[SandersonTankard, dateOfBirth=07-April-1992, tuitionFees=5000]
Course{ title=Course2, stream=Go, type=PartTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Sanderson Tankard has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Student[TeadorMiddup, dateOfBirth=27-November-1995, tuitionFees=5000]
Course{ title=Course1, stream=MySQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Teador Middup has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}

Assignment Menu
Please choose how would you like to proceed
1: Create manually assignment.
2: Show all assignments.
3: Choose course for assignment.
4: Give date and show students with all relevant student assignments.
5: Return.
Enter a number to proceed : |
```

PrivateSchoolSA

A list has appeared and we can see that with the date that we gave we can see the whole week here is 2021-01-04 to 2021-01-10 and we see all the students and their assignment during that week.

```

Student[TeadorMiddup, dateOfBirth=27-November-1995, tuitionFees=5000]
Course{ title=Course1, stream=MySQL, type=FullTime, startDate=01-January-2021, endDate=31-May-2021}
During period 2021-01-04 to 2021-01-10
Teador Middup has to deliver the assignments bellow.
Assignment{title=Assignment1, description=Create an Assignment1, subDateTime=08-January-2021, oralMark=0, totalMark=0}
```

```

Assignment Menu
Please choose how would you like to proceed
```

If we see more carefully this is true because we typed 07/01/2021 and we have as an output the right week 2021-01-04 to 2021-01-10 (07/01/2021 is inside this week) and in the assignment we can see for sure that the SubaDateTime for assignment belongs to that week 08/01/2021(08-january-2021).