PROJECT NAME: CLASS TEST ROUTINE AND SEAT PLAN MANAGEMENT SYSTEM

SUBMITTED BY: ABHIK BHATTACHARJEE – 1505040

BISHAL BASAK PAPAN – 1505043

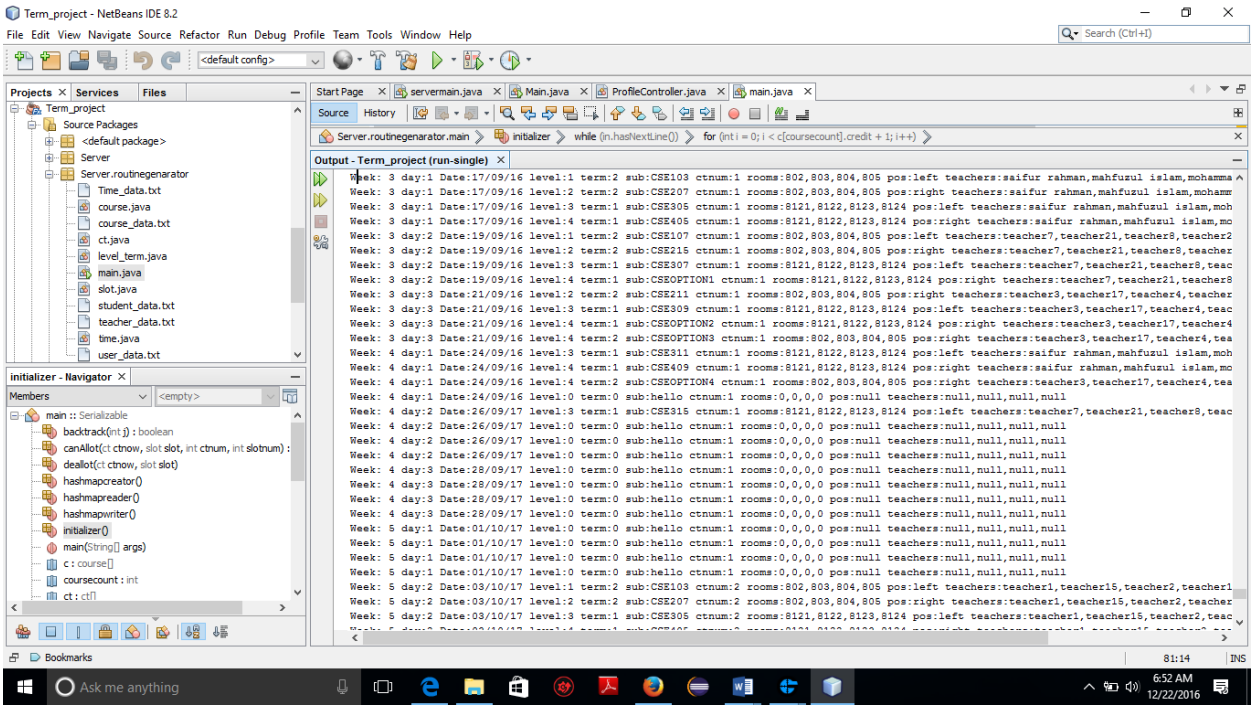
PROJECT SUPERVISOR: MD. SAIFUR RAHMAN

Available Features:

1) Code to generate the entire routine of all the departmental subjects of all the batches in a running term:

According to the constraints, there are three available days in a week to hold a class test: Saturday, Monday and Wednesday. Moreover, there must be a gap of at least two weeks between the class tests of the same subject of a level and term. The class tests will start on the third week of the term and all the class tests will finish before the 13th week. For holding a CT of a level-term, there must be 4 vacant rooms. As 2 different level-term's students can sit in one room and there are 8 available rooms, as much as 4 class tests of different level and term can be allotted in a single day. A student of any level and term will sit in the same place of the same room in this entire term. Maintaining all these constraints, using Backtracking algorithm, we have generated a possible routine along with the room no. of all the running batches in a level and term in the "main.java" class file in the "routinegenarator" sub package of the "Server" package. If it becomes impossible to allot all the subjects under these constraints, the program will tell that it is impossible to make a routine. The available dates (from the 3rd week to the 12th week) will be taken as input from the file named "time_data.txt". All the departmental courses and their credits along with the corresponding level and term will be taken as input from the file named "course_data.txt". There are instructions on how to put data in the corresponding files in "Read_me.txt" file. All these text files will be in the server computer.

Sample routine as output:

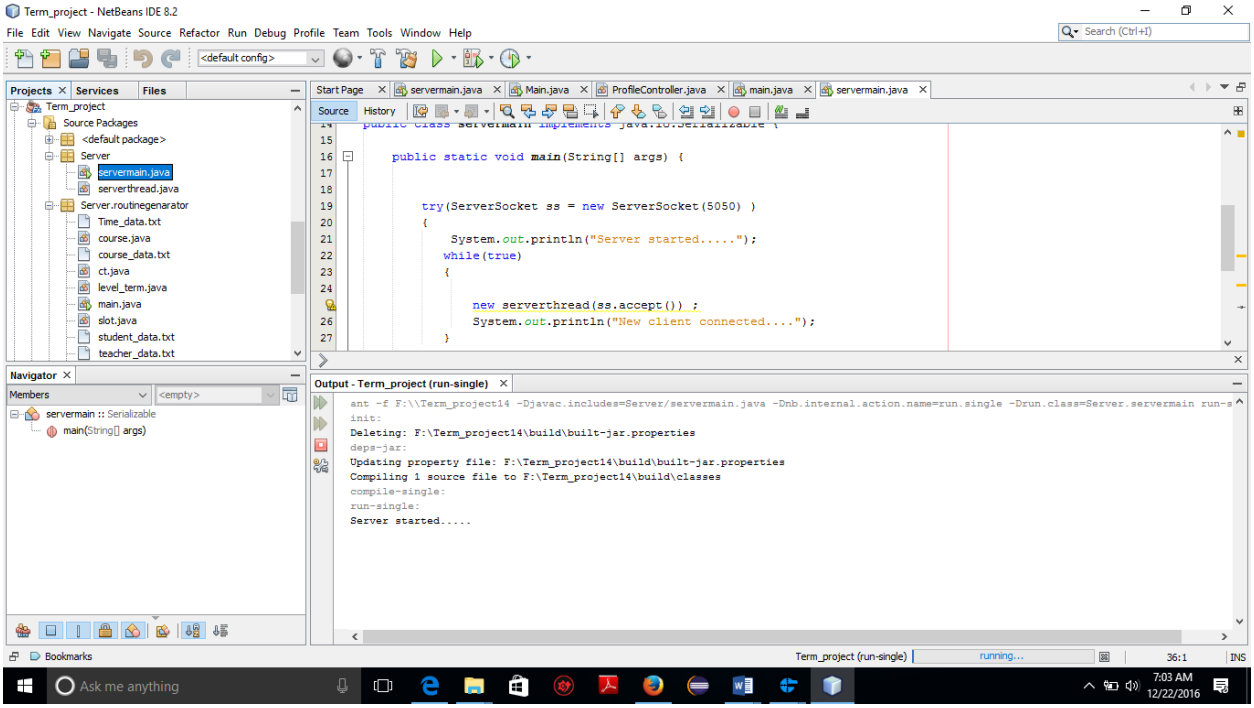


2) <u>Any number of clients can connect to the server:</u>

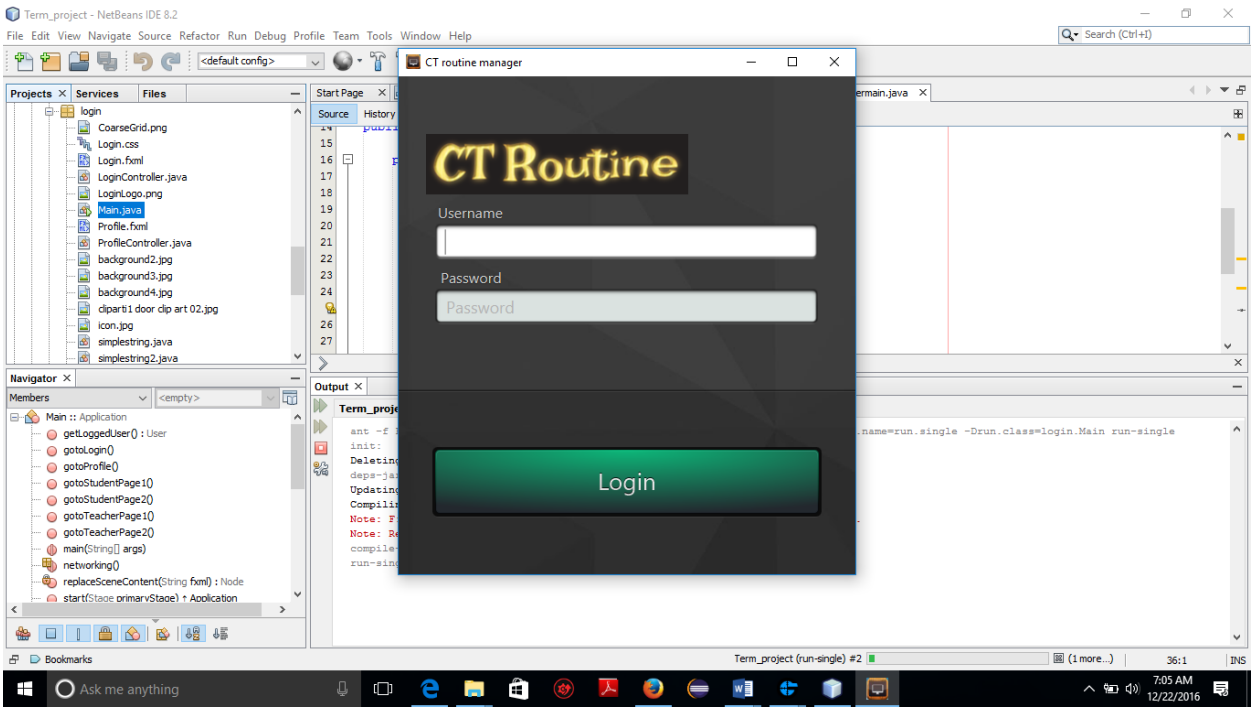
If the above mentioned “main.java” class file is run in a computer and then the “servermain.java” class file in the “Server” package is run in that computer, the computer will start working as a server. The “servermain.java” class file will show “Server started…” as output. According to our program, if the server remains started, then any number of clients can connect to the server. **But if the “servermain.java” class file is not in a running state, no client can connect to it. So to allow the clients to be connected, the “servermain.java” class file must be in a running state.** To connect to the server, the “Main.java” class file in the “login” package must be run in the client computer. If a client can connect to the server, then the “Login” interface will appear in his computer and servermain class will show that a client is connected.

An important point to be noted that we have used the local host name "127.0.0.1" as server in the "Main.java" class of "login" package. So for further use, the local host name must be as the name/ip of the server computer.

After the servermain.java class file is run:

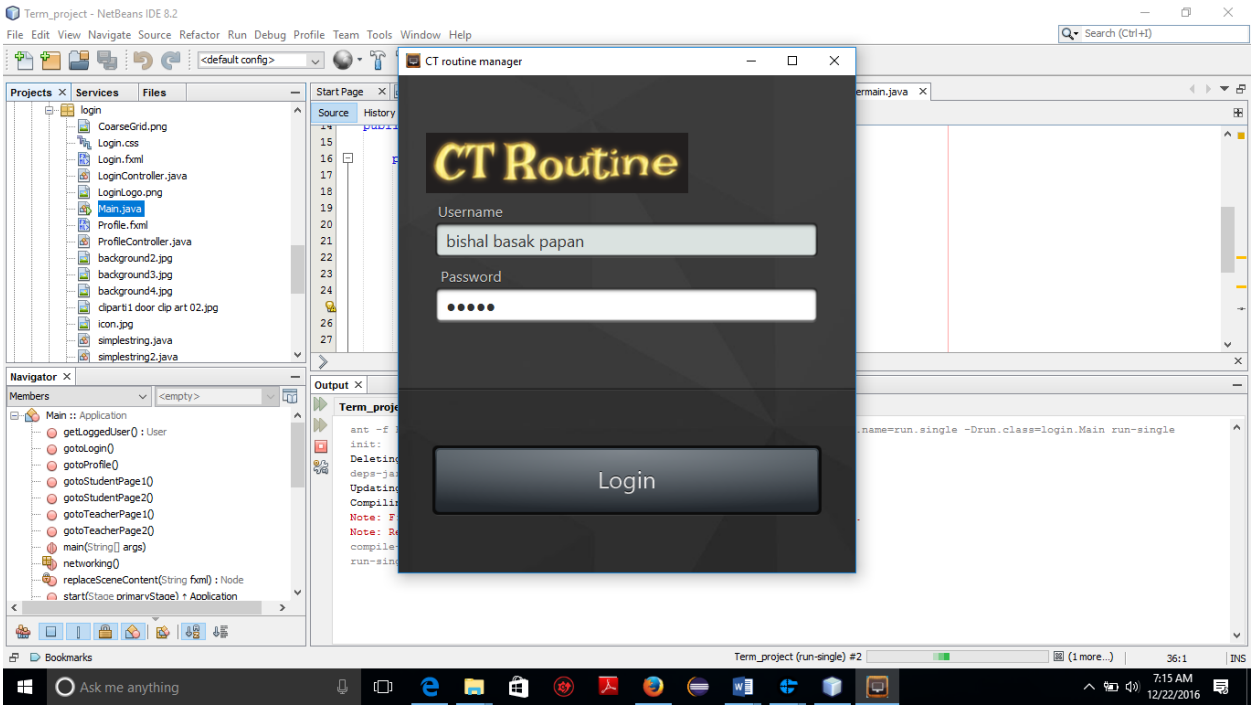


After a client connects to the server successfully he will see this interface:



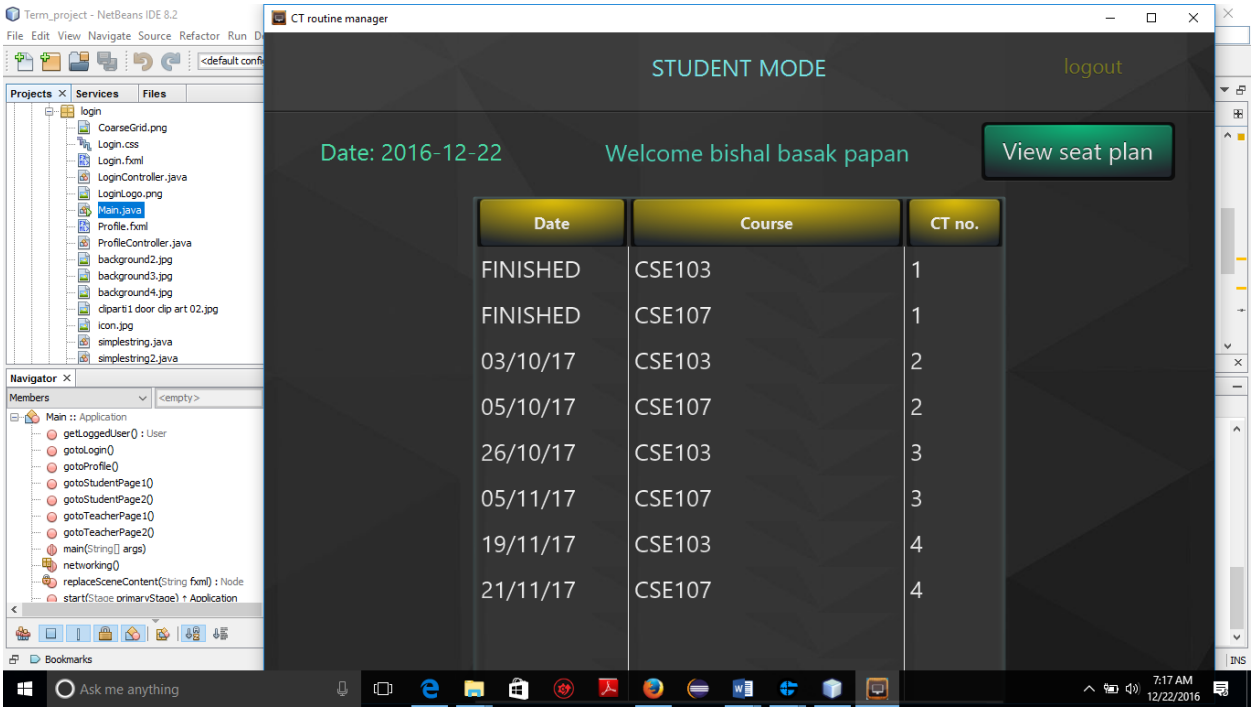
3) <u>Two different interfaces according to the username and password:</u>

In the login interface, the client will put his username and password in the corresponding text boxes. After pressing the “Login” button or pressing “Enter” key, if the username and password matches with a username and password of a student in “user_data.txt” file, then he will only have access to the “Student Mode” of this software.

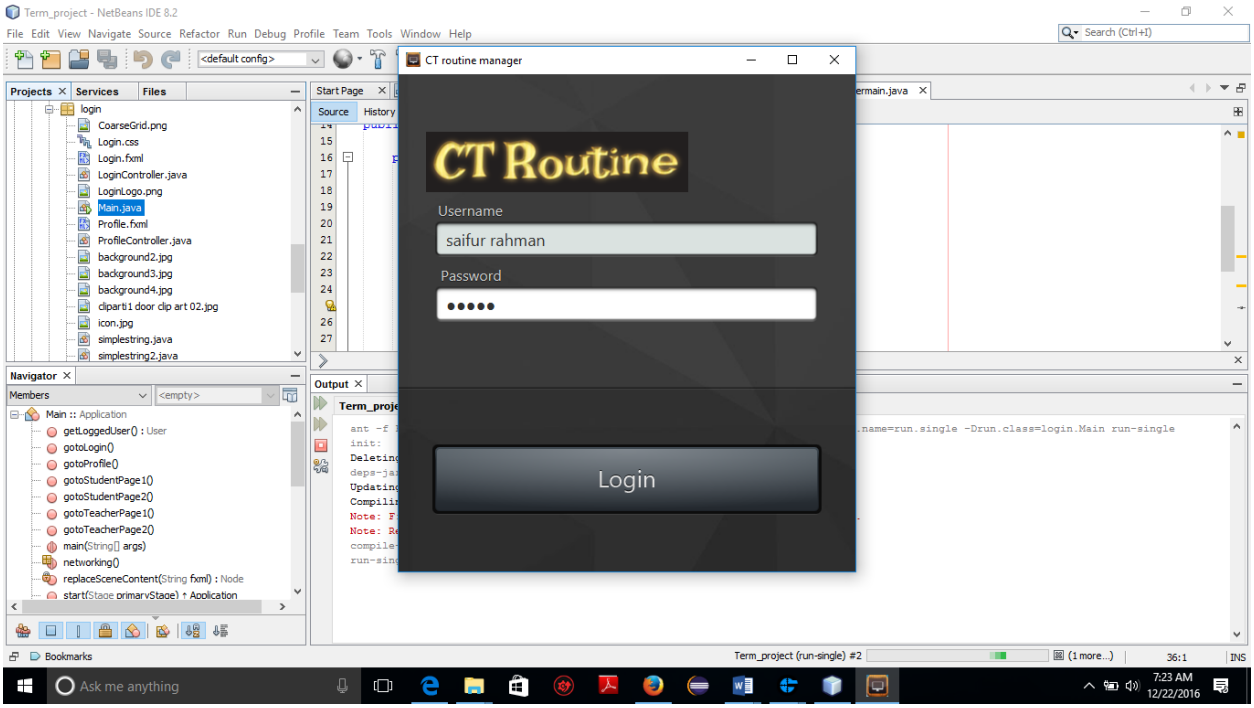


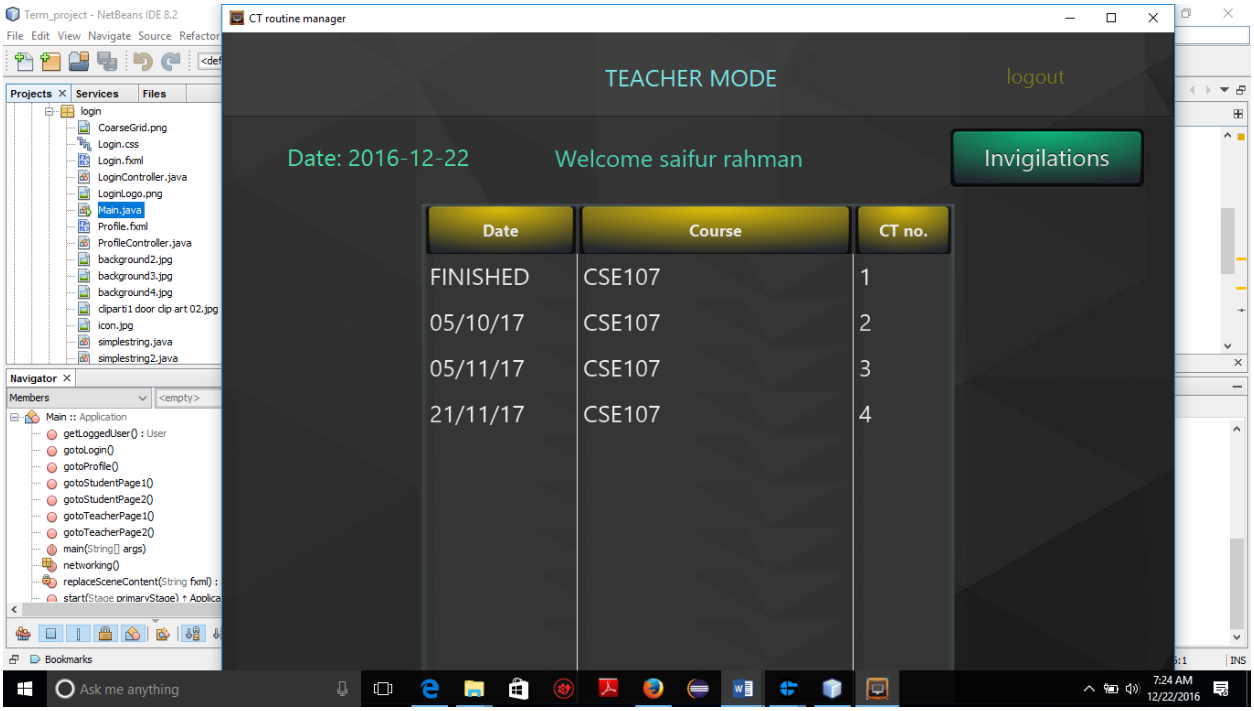
The “Login” button will change its color if the cursor comes on the button.

After logging in, the following interface will appear before him.



But if the username and password matches with the username and password of a teacher, then he will have access to the “Teacher mode”:

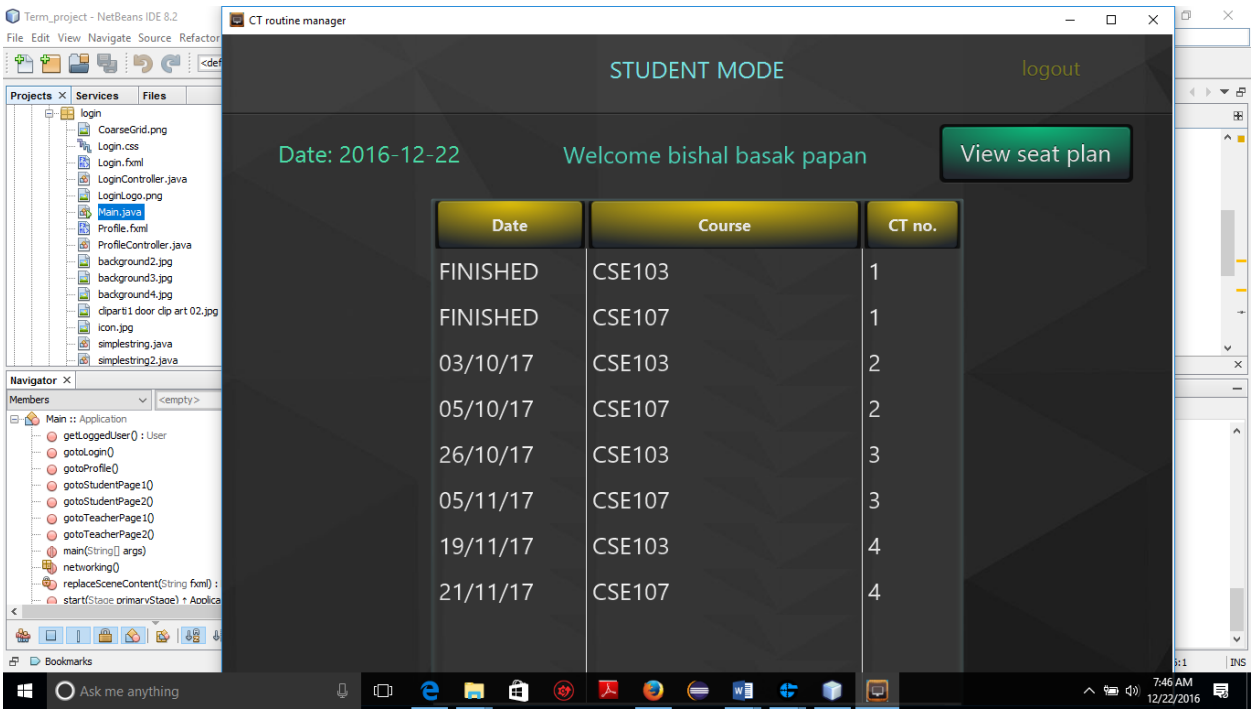




Both the student and the teacher will see the current date (in “yyyy-mm-dd” format) in the top left corner of the corresponding interfaces.

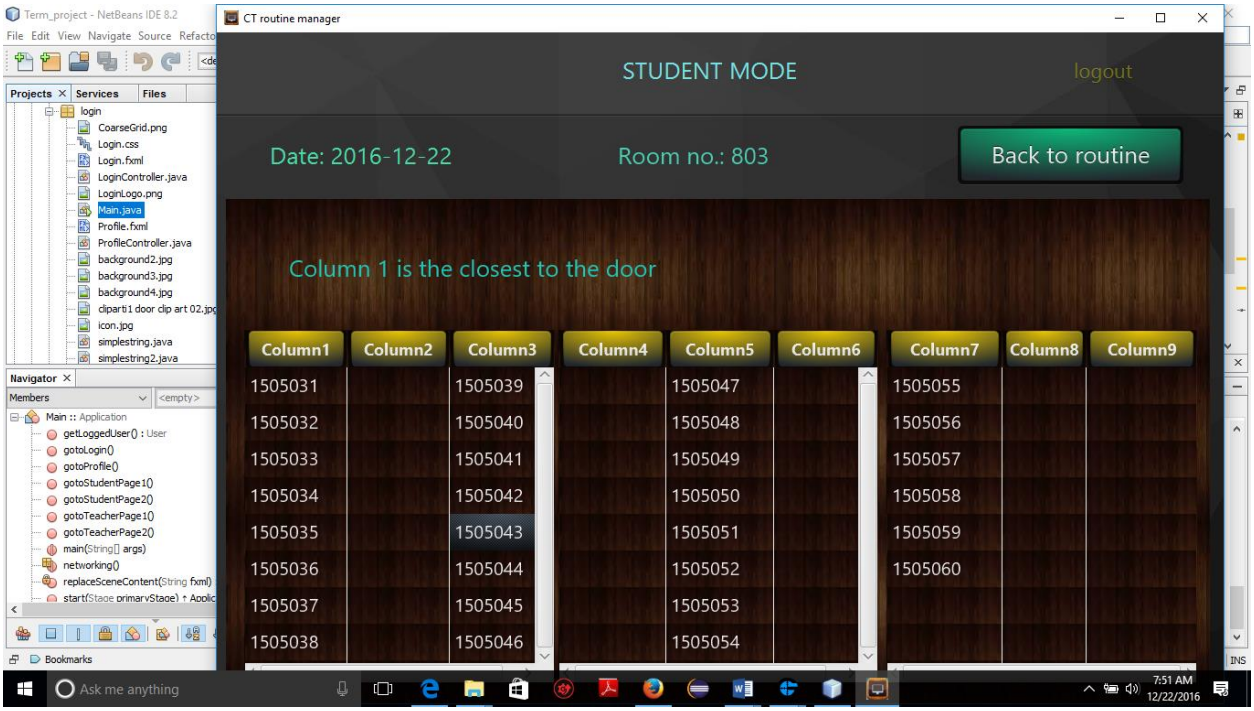
4) A student will see his courses’ CT dates and his seat:

A student will not be able to see the entire Class Test routine of the term. Rather he will only be able to see the Class Test routine of his courses. The corresponding level-term and roll number of a student will be in the “student_data.txt” file. The class test routine will be derived from the output of the routine generator code. This routine will be sorted by date. If a class test is finished, then the corresponding “Date” cell of that class test will show “FINISHED”. Thus the student will be able to know about his very next CT. (In our program, we have given some dates of the year 2016 and some dates of the year 2017 in “Time_data.txt” file as a sample)



From the routine, a student can know the date and number of Class Test of a course. But he cannot know the invigilator's name.

If he presses the "View seat plan" button, he will be able to see in which room and in exactly which place he has to sit for all the class tests. The room number will be shown in the top middle portion of the interface. His roll number will be highlighted so that he can easily find his seat.



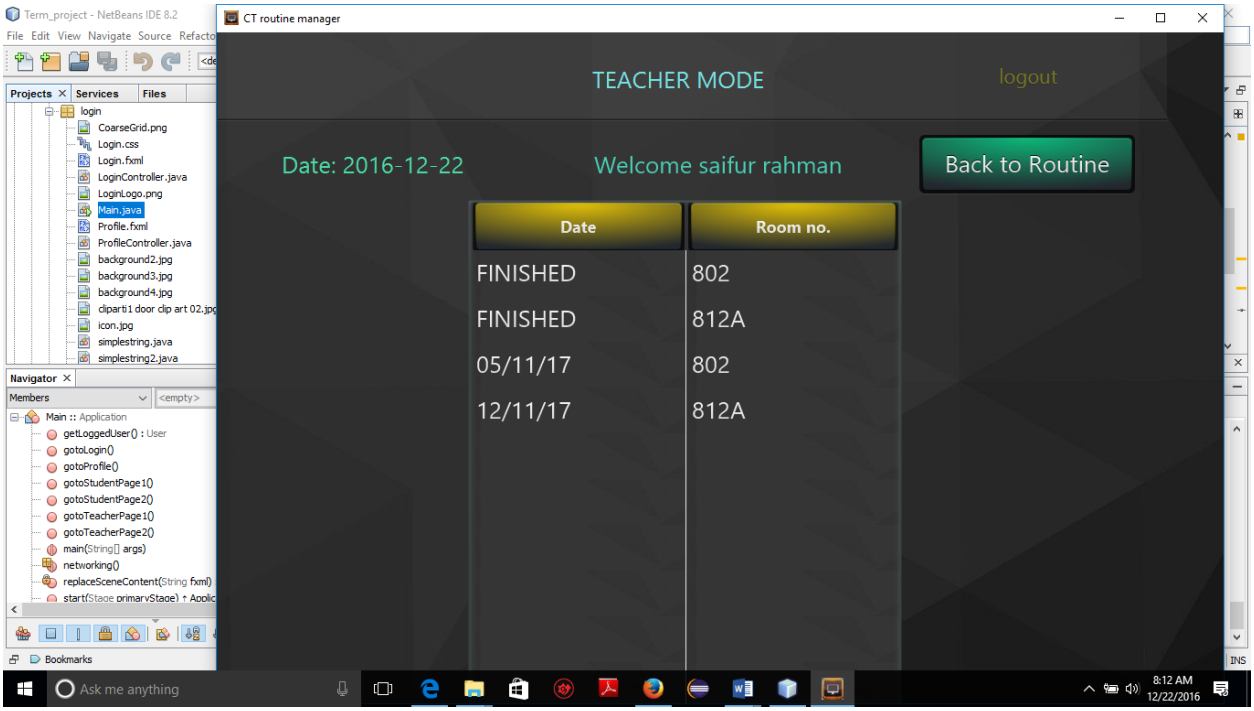
He can go back to the routine page by pressing "Back to routine" button and logout from the software by clicking on the "logout" hyperlink.

5) A teacher will see his courses' CT dates and in which rooms he has to invigilate:

The courses under a teacher have to be given in the "teacher_data.txt" file. According to the information in this file and the output of the routine generator program, a teacher will only be able to see the dates and numbers of the class

tests of his courses in “Teacher mode”. Just like in the student mode, the teacher will also see “FINISHED” in the date cell of a CT if the CT is finished.

If he presses the “Invigilations” button, he will be able to see in which rooms and in which dates he has to act as an invigilator too.



Some improvements that can be made in the project:

1) According to our software, a teacher cannot see the entire seat plan of the room he has to invigilate. He can only know the room number. But it would be greatly helpful for the teacher if he could see the seat plan. So the feature of visualizing the seat plan by a teacher can be added to the software.

2) The feature of showing the results of the previous Class Tests to the corresponding students and teachers can also be added. This feature will surely improve the project to a great extent.

3) This software would be more user friendly if an error message was shown in the login page if username or password or both does not match with any of the client information. In our software, there is no system of showing such error message but if any client gives the correct username and wrong password, he will get an error message. Moreover, after making mistake, if

any client puts the correct username and password, he will be able to log in.

4) A notification bar showing the next class test can also be shown in both of the modes of the software to make it more useful.