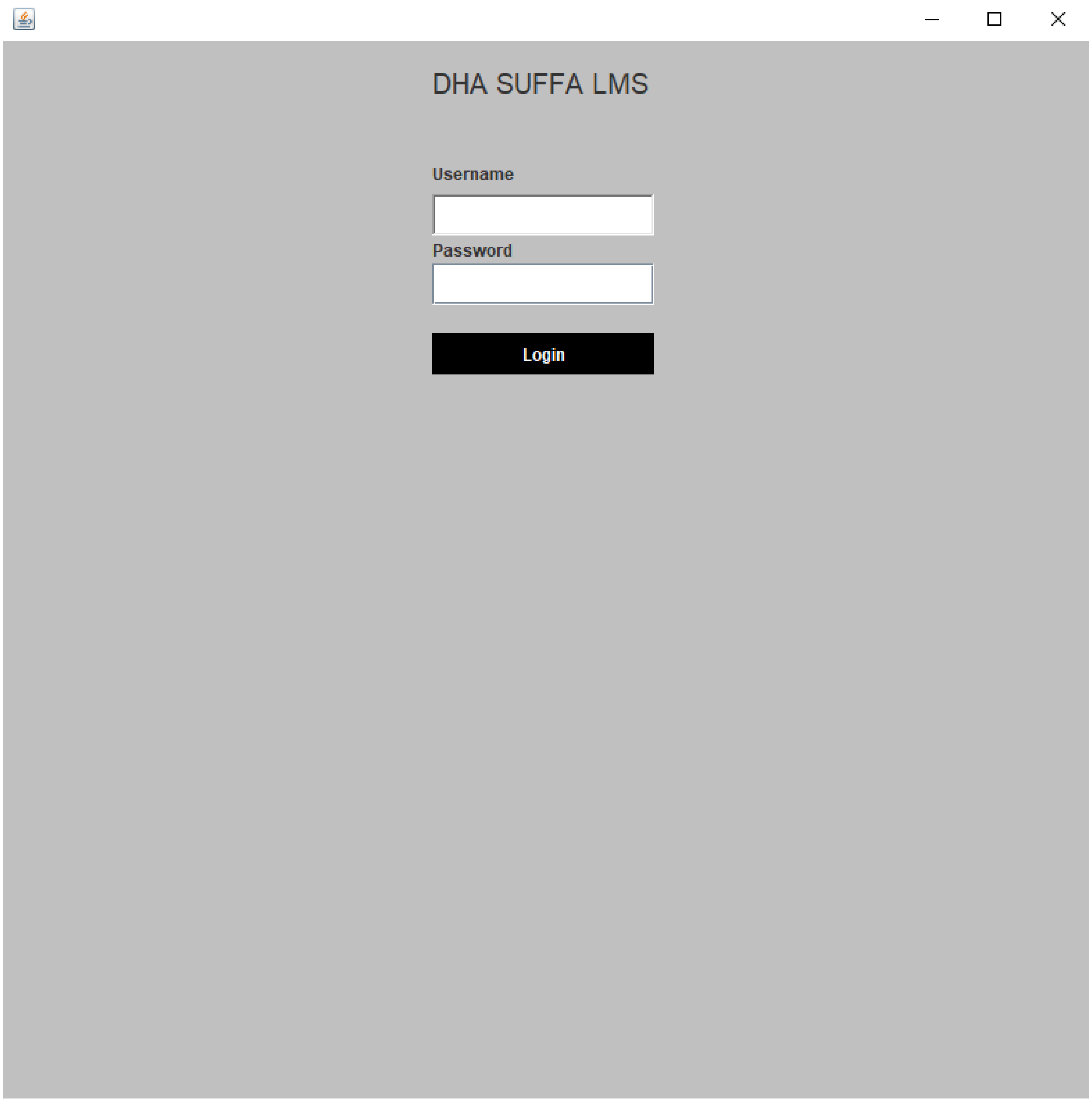
Name: Ali Usaid

Roll no: SE221025.

Object-Oriented Programming – Project

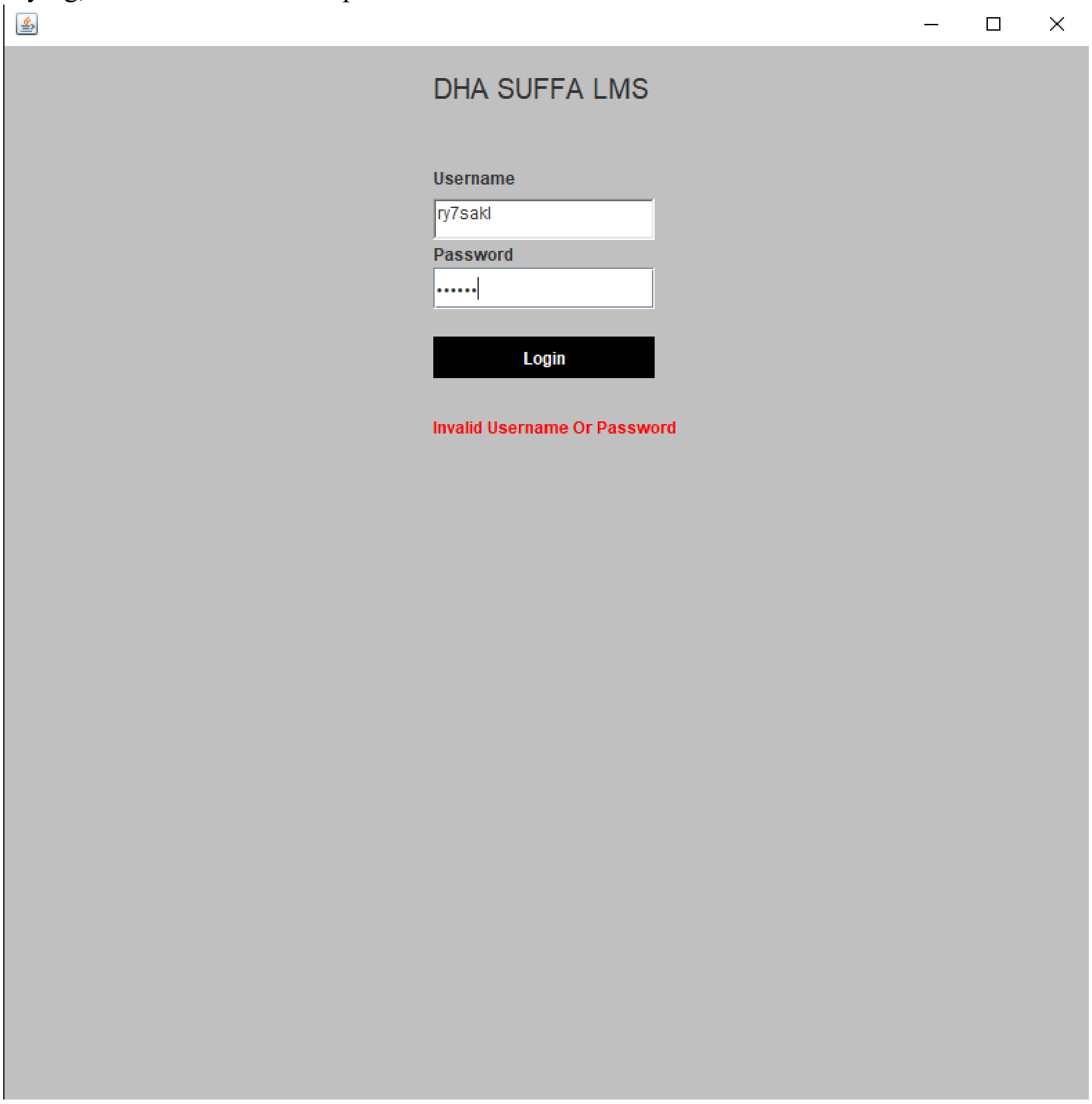
The project is GUI based on the Java Swing & AWT library the project contains 3 pillars of OOPS except for polymorphism it also contains aggregation and composition I have handled all the exceptions and wrong cases for example wrong password or username submitting a question to question bank but forget to choose MCQ or forget to input question or haven't selected a right answer making a quiz but haven't picked any questions so all type of these things have been covered in this project lets take a look at first screen when we start the project



When starting the project, if the initial screen fails to load properly on your system, you should minimize it and then reopen it again. To access the teacher Learning Management System (LMS), please use the following credentials:

- Username: OOPS
- Password: OOPS123

If the user enters an incorrect username or password, the system will display a message saying, "Invalid username or password."


DHA SUFFA LMS

Username

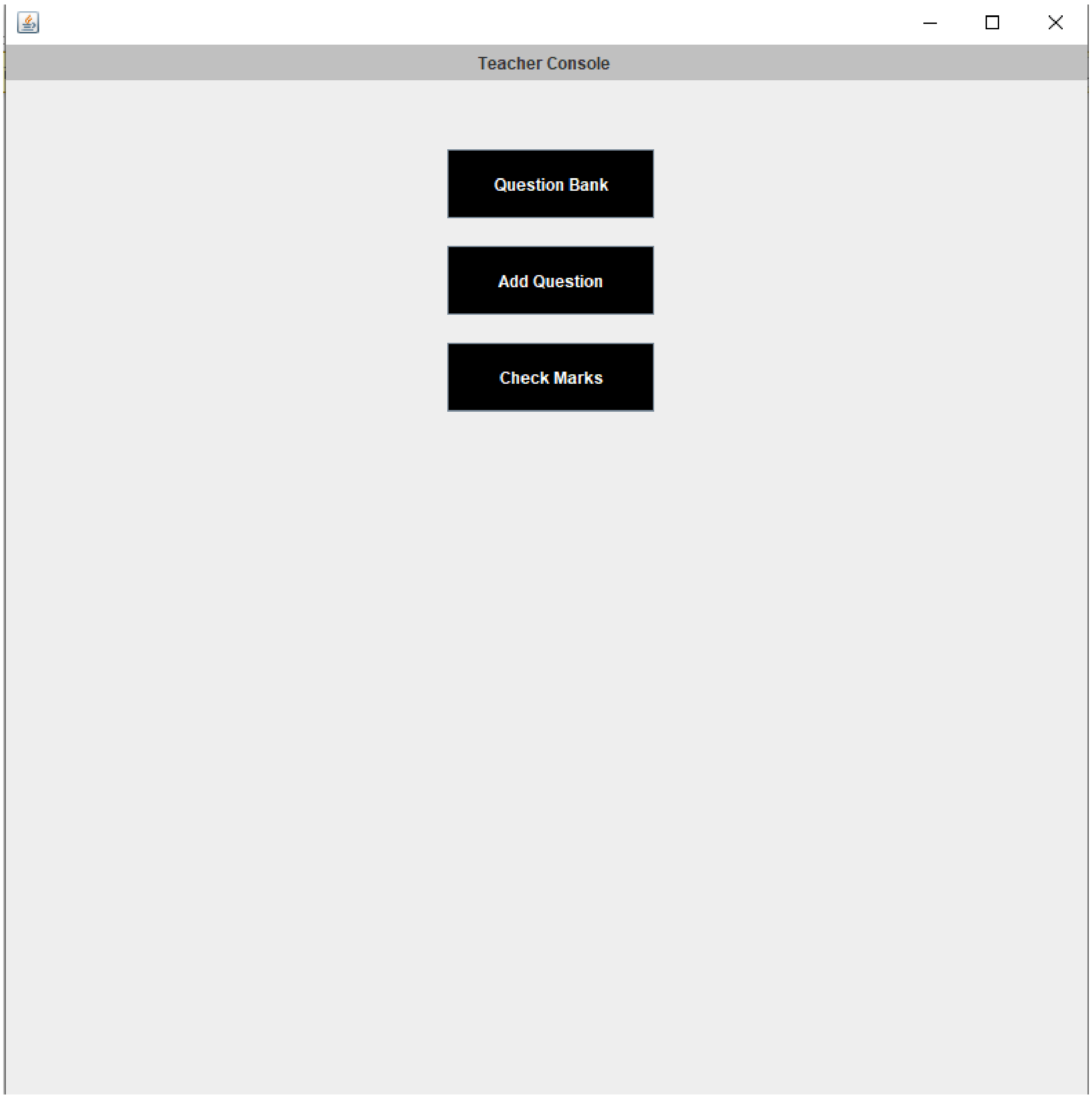
ry7sakl

Password

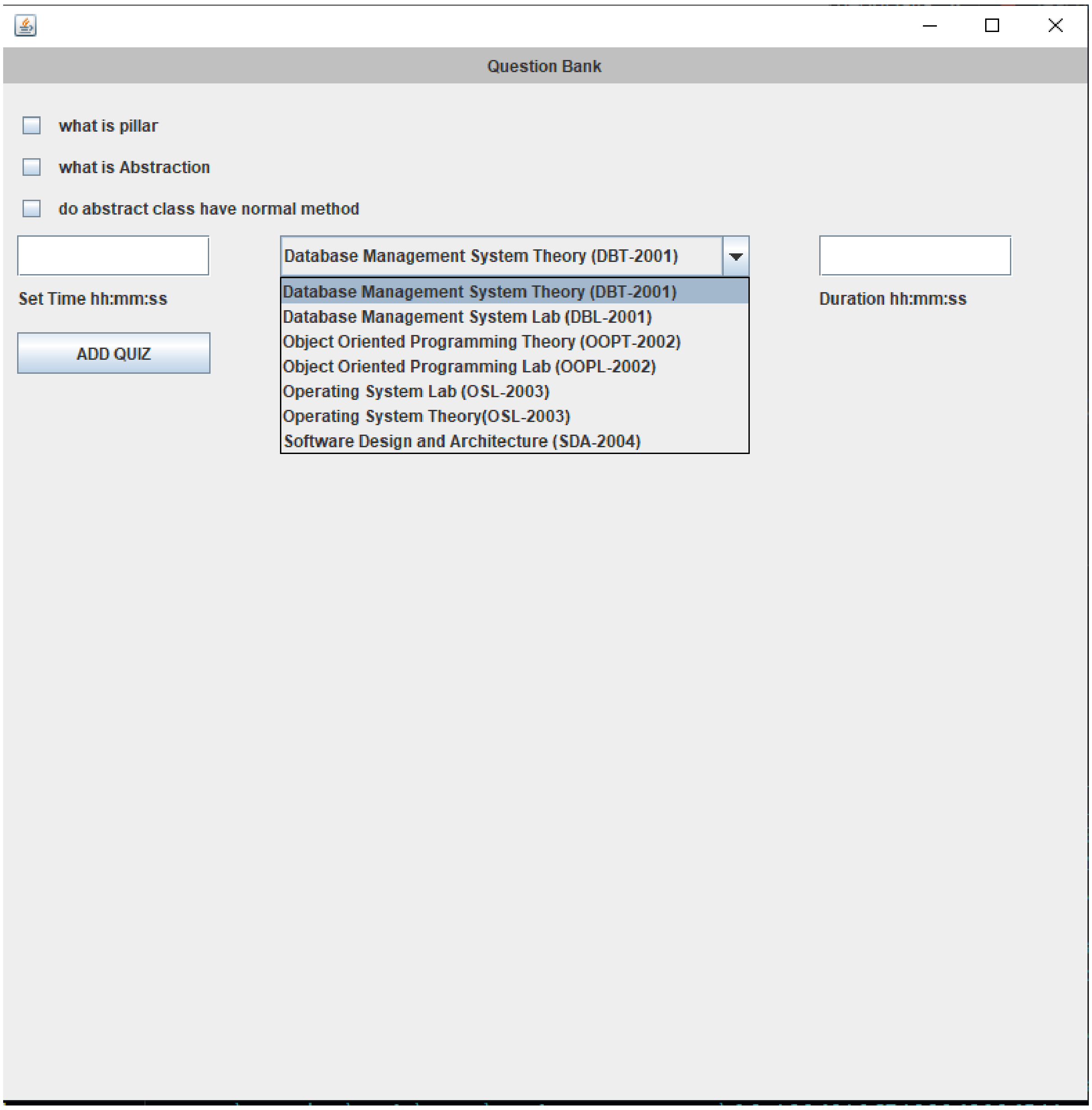
••••••

Invalid Username Or Password

After entering the correct username and password, you will be able to access the teacher console, which will have the following appearance.



As we can see, there are three buttons on the screen. The first button leads you to the question bank, where the teacher can generate a quiz at a specific time of the current day. The teacher will also specify the course name and duration of the quiz. Once generated, the quiz will be saved in a file. When the local time of the student's system matches the specified time for the quiz, the quiz button will disappear from the student console.


Question Bank

- [ ] what is pillar
- [ ] what is Abstraction
- [ ] do abstract class have normal method

Set Time hh:mm:ss

Database Management System Theory (DBT-2001)

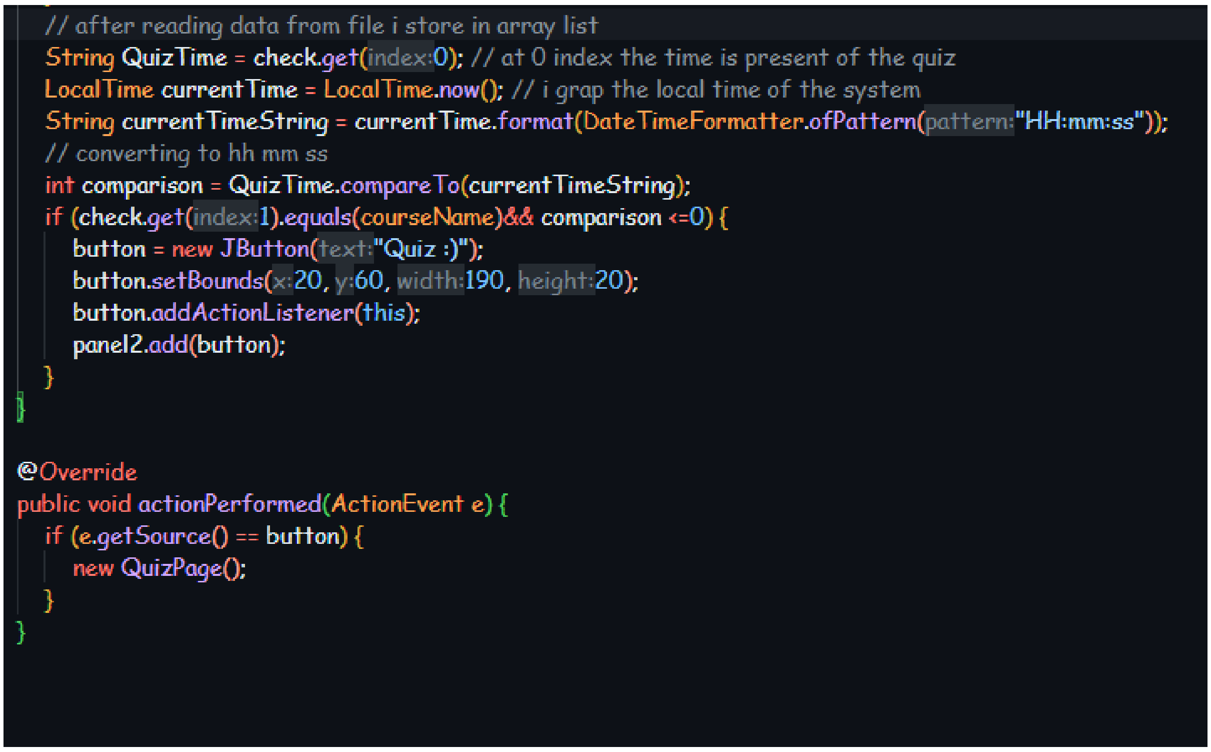
Database Management System Theory (DBT-2001)
Database Management System Lab (DBL-2001)
Object Oriented Programming Theory (OOPT-2002)
Object Oriented Programming Lab (OOPL-2002)
Operating System Lab (OSL-2003)
Operating System Theory(OSL-2003)
Software Design and Architecture (SDA-2004)

Duration hh:mm:ss

ADD QUIZ

The logic behind this is simple.

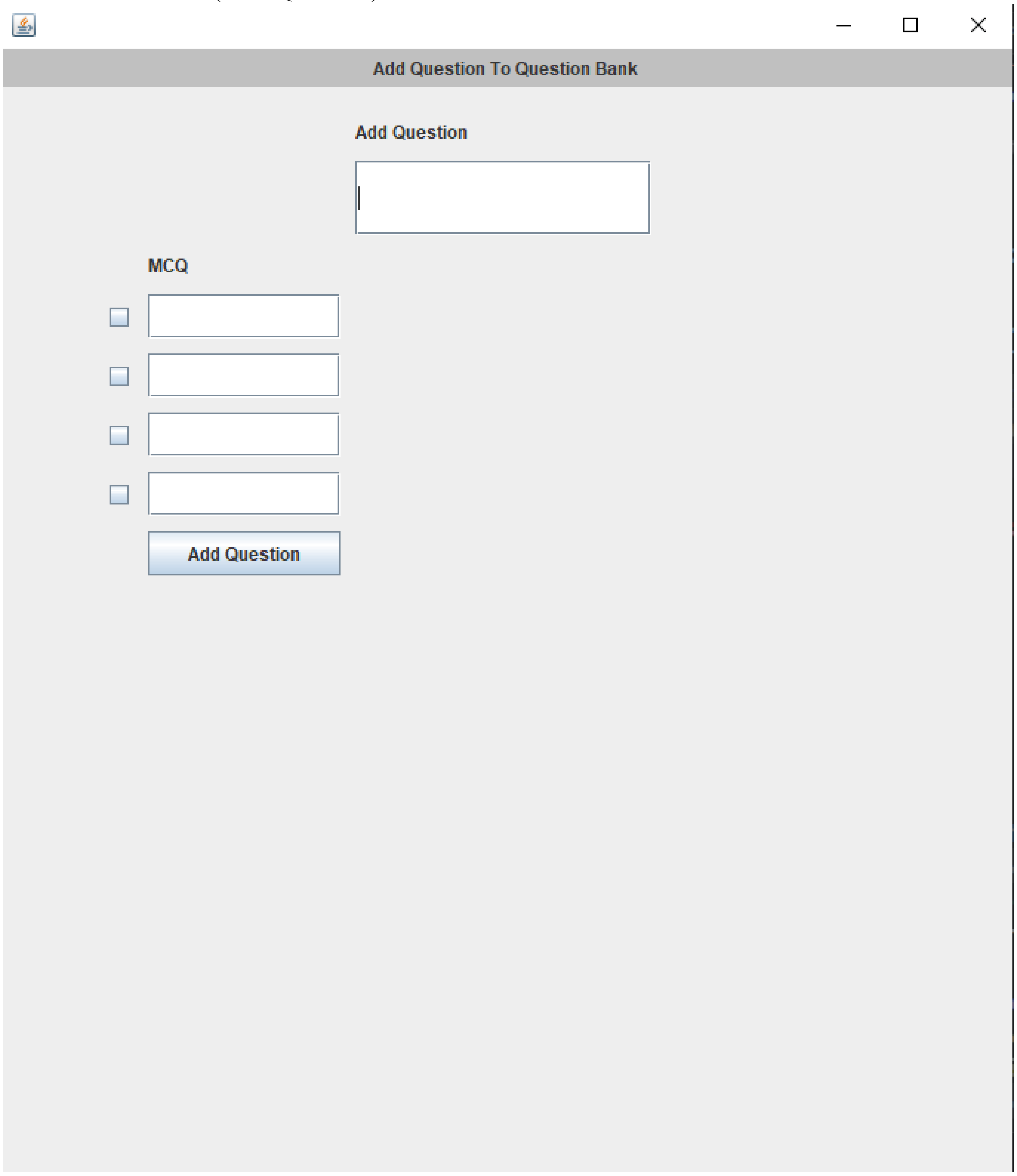
Let's take a closer look at the code



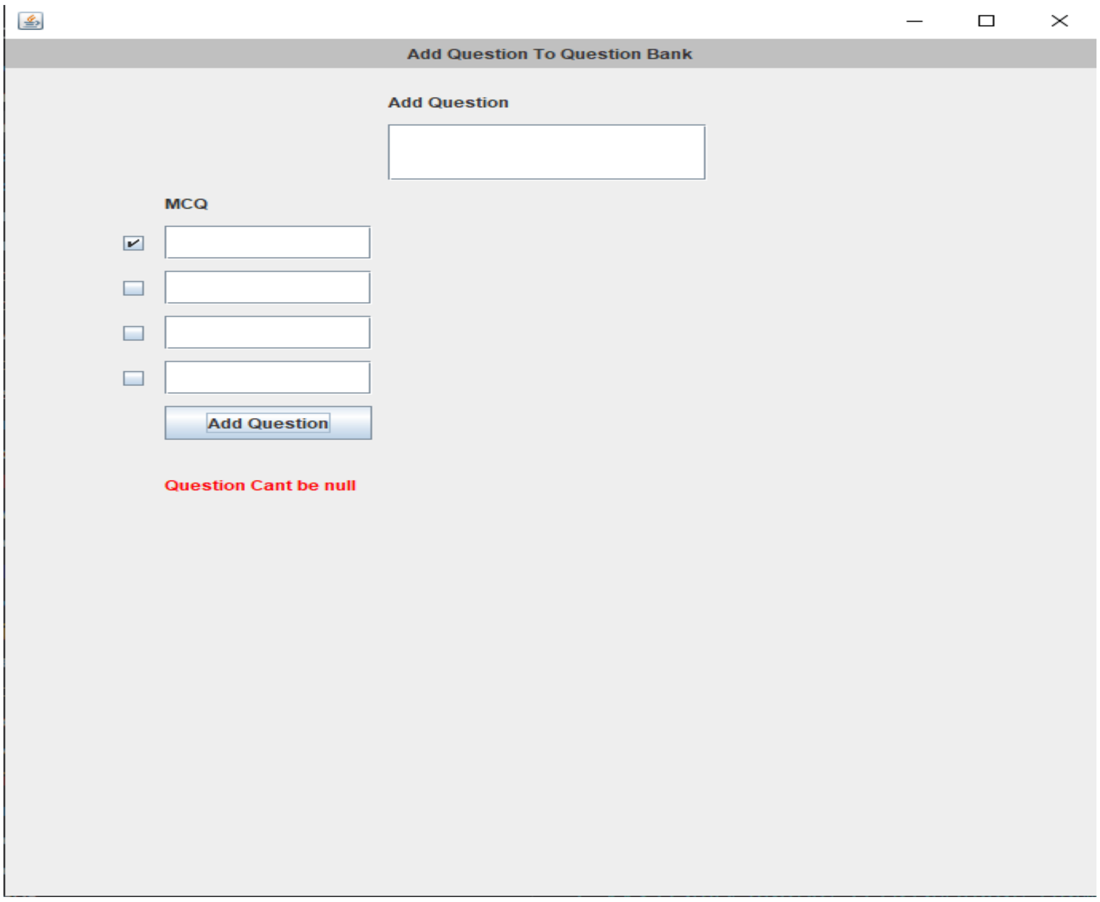
```
    // after reading data from file i store in array list
    String QuizTime = check.get(index:0); // at 0 index the time is present of the quiz
    LocalTime currentTime = LocalTime.now(); // i grap the local time of the system
    String currentTimeString = currentTime.format(DateTimeFormatter.ofPattern(pattern:"HH:mm:ss"));
    // converting to hh mm ss
    int comparison = QuizTime.compareTo(currentTimeString);
    if (check.get(index:1).equals(courseName)&& comparison <=0) {
        button = new JButton(text:"Quiz :)");
        button.setBounds(x:20, y:60, width:190, height:20);
        button.addActionListener(this);
        panel2.add(button);
    }
}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == button) {
        new QuizPage();
    }
}
```

First, we will grab the time of the quiz. Then, we will obtain the local time of the user and convert it into HH:MM:SS format. Next, we will compare the two times using an "if" condition. If the quiz time is greater than or equal to the local time, we will display the quiz button on the screen.

Now come back to (Add Question) screen

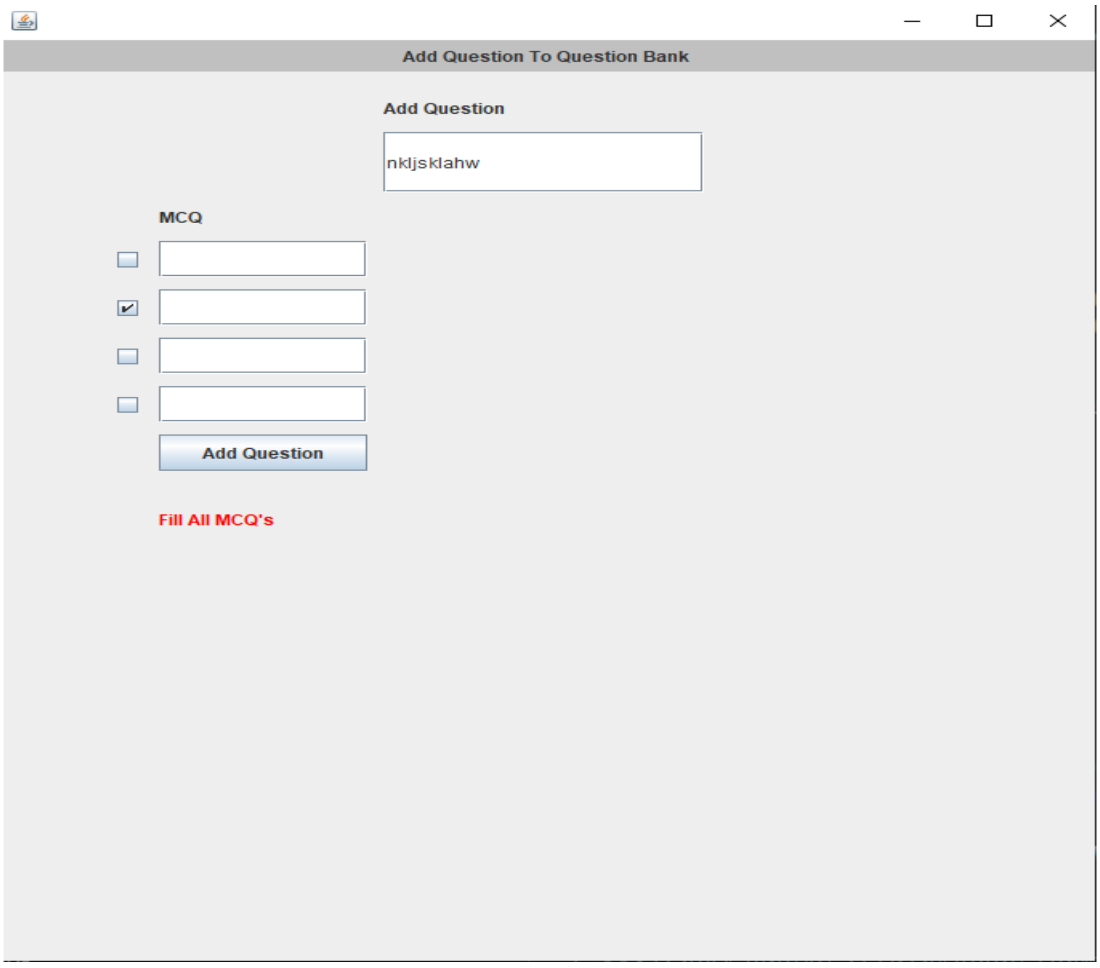


I have accounted for all the possible error cases. For instance, if the user forgets to input the multiple-choice question (MCQ), the question won't be saved in the file. Similarly, if the user fails to write a question, it will also be handled appropriately.



Question Cant be null

The question cannot be null. If the user leaves the question field empty, an error message will be displayed on the screen, and the question will not be saved in the file.


Add Question To Question Bank

Add Question

nkljsklahw

MCQ

☐

✓

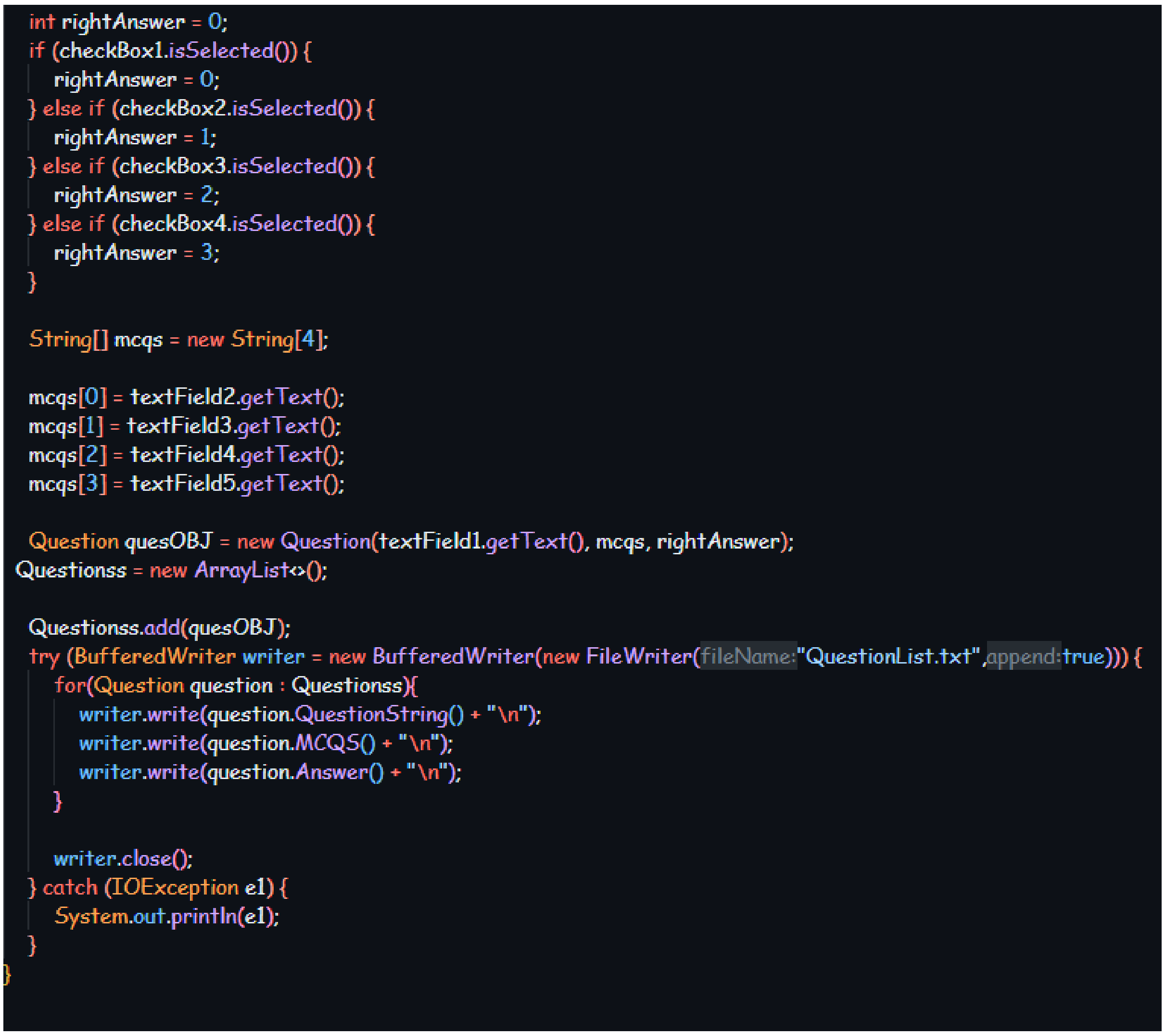
☐

☐

Add Question

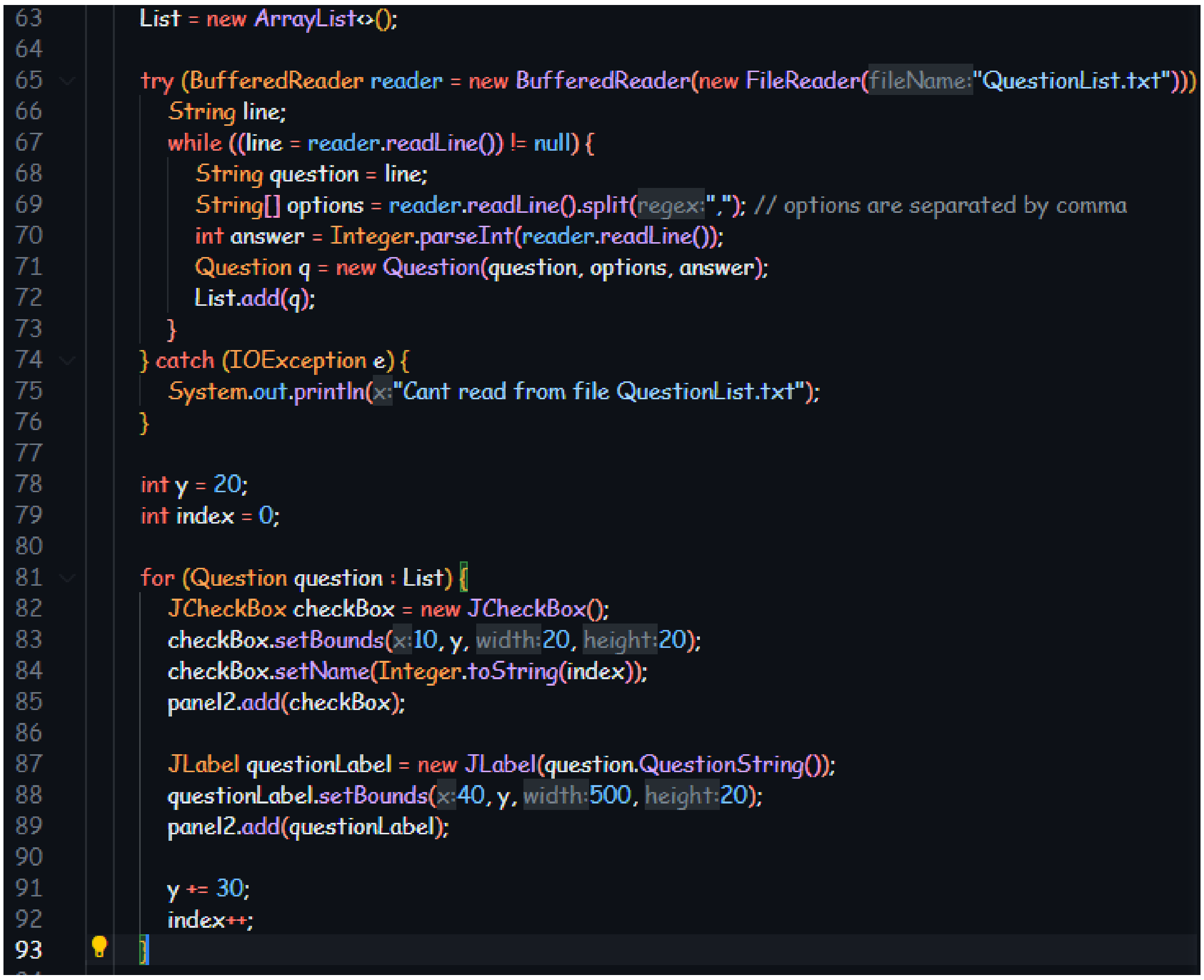
Fill All MCQ's

If the user leaves the MCQ field empty, an error message will be displayed, prompting them to fill in all the MCQs. the message will serve as a reminder to complete all the multiple-choice questions before proceeding. By providing clear instructions, we can ensure that users understand the requirement and take appropriate action to input the necessary information.



```java
  int rightAnswer = 0;
  if (checkBox1.isSelected()) {
    rightAnswer = 0;
  } else if (checkBox2.isSelected()) {
    rightAnswer = 1;
  } else if (checkBox3.isSelected()) {
    rightAnswer = 2;
  } else if (checkBox4.isSelected()) {
    rightAnswer = 3;
  }

  String[] mcqs = new String[4];

  mcqs[0] = textField2.getText();
  mcqs[1] = textField3.getText();
  mcqs[2] = textField4.getText();
  mcqs[3] = textField5.getText();

  Question quesOBJ = new Question(textField1.getText(), mcqs, rightAnswer);
 Questionss = new ArrayList<>();

  Questionss.add(quesOBJ);
  try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName:"QuestionList.txt",append:true))) {
    for(Question question : Questionss){
      writer.write(question.QuestionString() + "\n");
      writer.write(question.MCQS() + "\n");
      writer.write(question.Answer() + "\n");
    }

    writer.close();
  } catch (IOException e1) {
    System.out.println(e1);
  }
}
```

First, I will check which checkbox is selected using an if-else statement. Then, I will retrieve all the MCQs from the text field. Next, I will create an object of the Question class and initialize an Array List to store the questions. I will add the object to the Array List. To write the question to a file, I will use a Buffered Writer, ensuring that the file is in append mode. This way, multiple questions can be stored in the file. When displaying the question bank screen, the questions will be retrieved and displayed accordingly. It's a straightforward process that allows for easy management of the questions.



```java
63          List = new ArrayList<>();
64  
65          try (BufferedReader reader = new BufferedReader(new FileReader(fileName:"QuestionList.txt")))
66              String line;
67              while ((line = reader.readLine()) != null) {
68                  String question = line;
69                  String[] options = reader.readLine().split(regex:","); // options are separated by comma
70                  int answer = Integer.parseInt(reader.readLine());
71                  Question q = new Question(question, options, answer);
72                  List.add(q);
73              }
74          } catch (IOException e) {
75              System.out.println(x:"Cant read from file QuestionList.txt");
76          }
77  
78          int y = 20;
79          int index = 0;
80  
81          for (Question question : List) {
82              JCheckBox checkBox = new JCheckBox();
83              checkBox.setBounds(x:10, y, width:20, height:20);
84              checkBox.setName(Integer.toString(index));
85              panel2.add(checkBox);
86  
87              JLabel questionLabel = new JLabel(question.QuestionString());
88              questionLabel.setBounds(x:40, y, width:500, height:20);
89              panel2.add(questionLabel);
90  
91              y += 30;
92              index++;
93          }
```

The logic used in the question bank to retrieve data from the file where all the questions have been added may seem complicated at first, but it's actually quite simple. Here's a breakdown of the logic:

- First, an Array List called "list" is initialized to store the questions.
- Reading from the file, a variable named "line" is declared.
- The while loop condition is set as (line = reader.readLine()) != null. This approach is used to prevent skipping the first line of the file. By reading the line and checking if it is not null, it ensures that the loop executes if there are more lines to read.
- Inside the loop, the question is retrieved from the file.
- In line 69, the MCQs are obtained by splitting the line using the split method. The MCQs are stored in a String array.
- Line 3 of the file represents the correct answer.

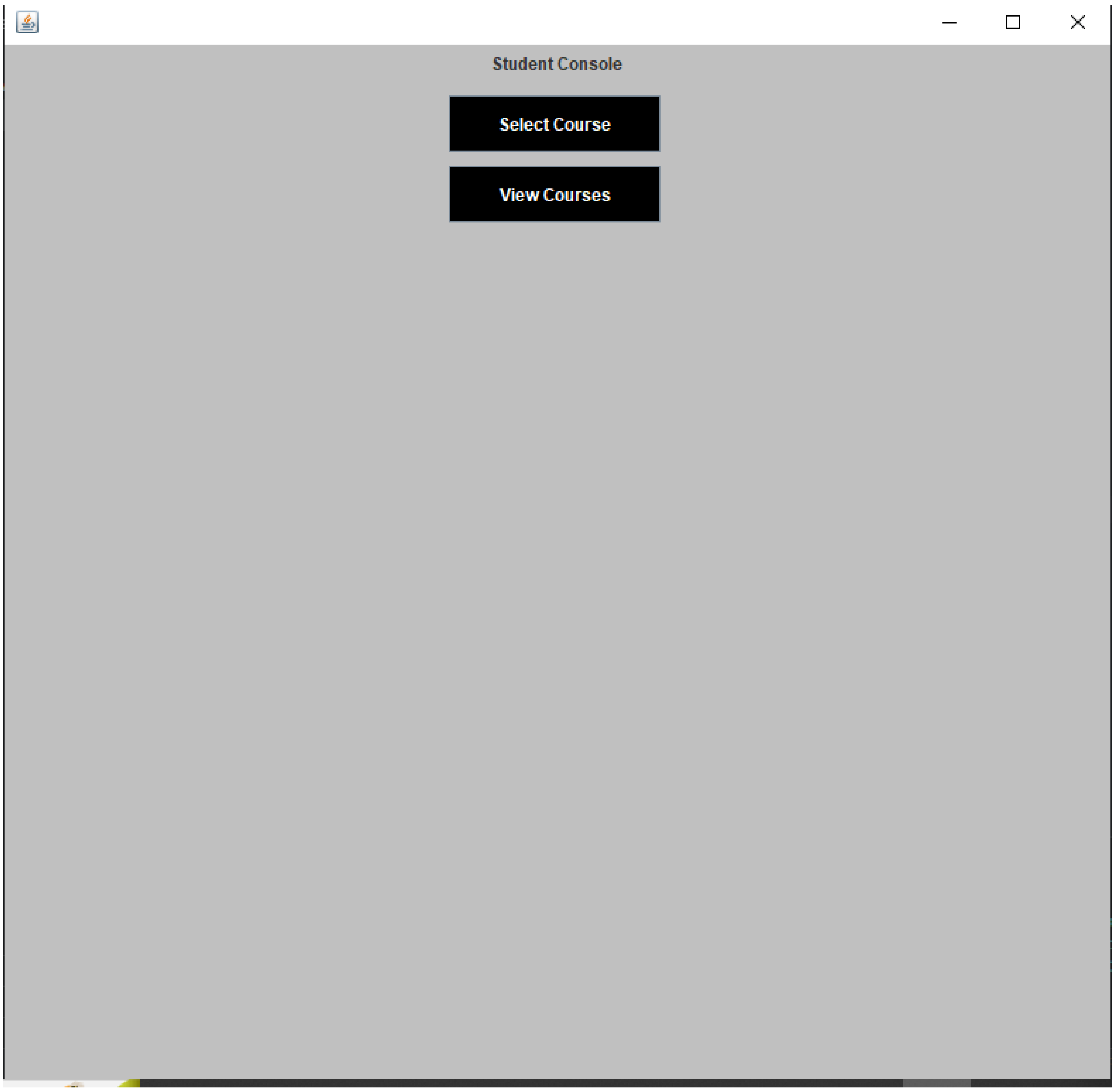
To summarize, logic involves reading the lines from the file one by one, processing the data, and storing it in the variables or data structures for later use. By understanding the purpose and flow of each step, it becomes easier to comprehend the logic behind the code.

This is all for the Teacher Screen there was another button Check marks which will display the marks of the student who attempt the quiz that page is really simple so I don't want to go in-depth of it now we will see Student Console

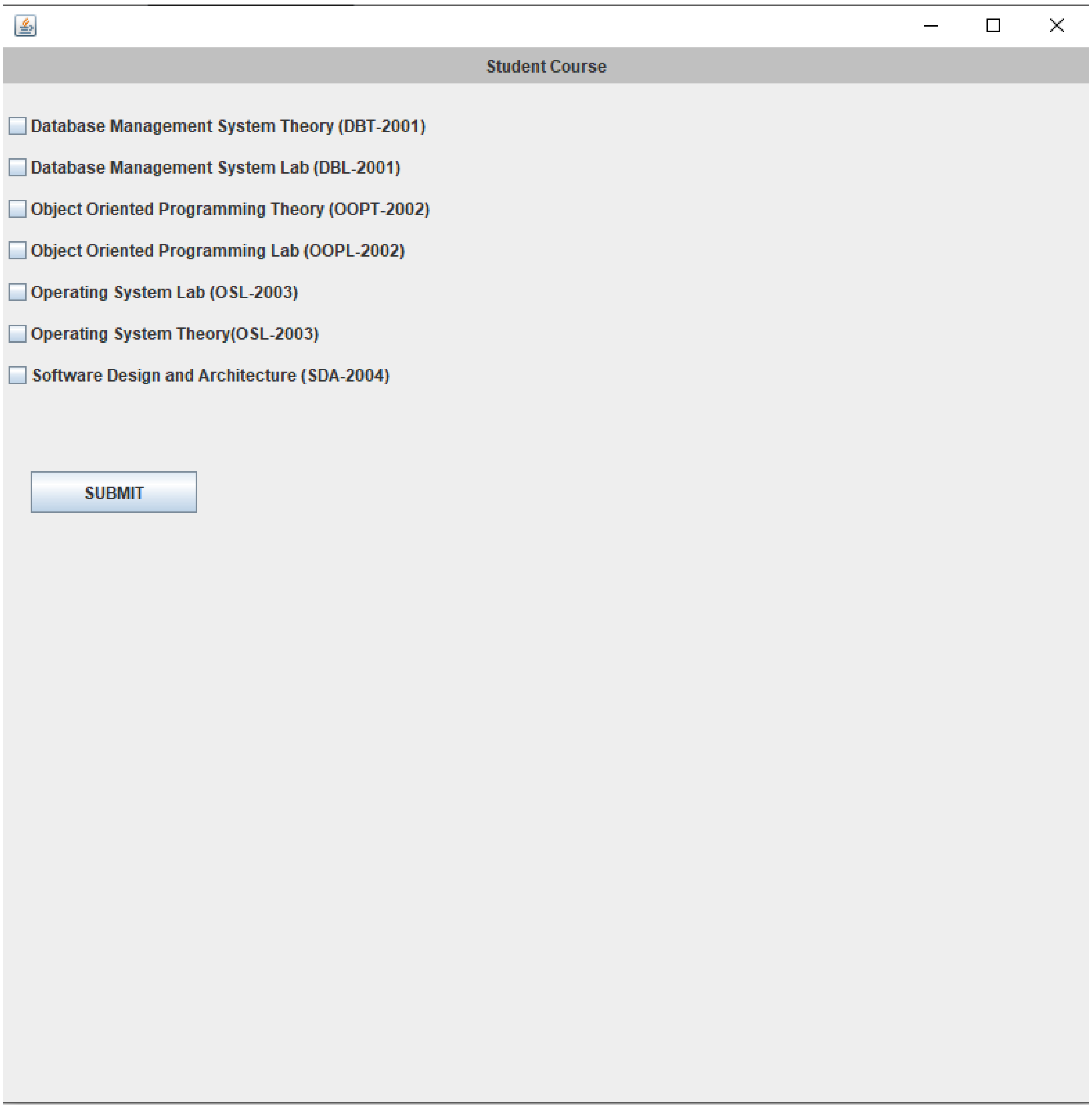
To login student Console please use the following credentials.

- Username: Ali Osaid
- Password: TAHMYVES

After entering the correct password you will log in to student LMS and it will look like this

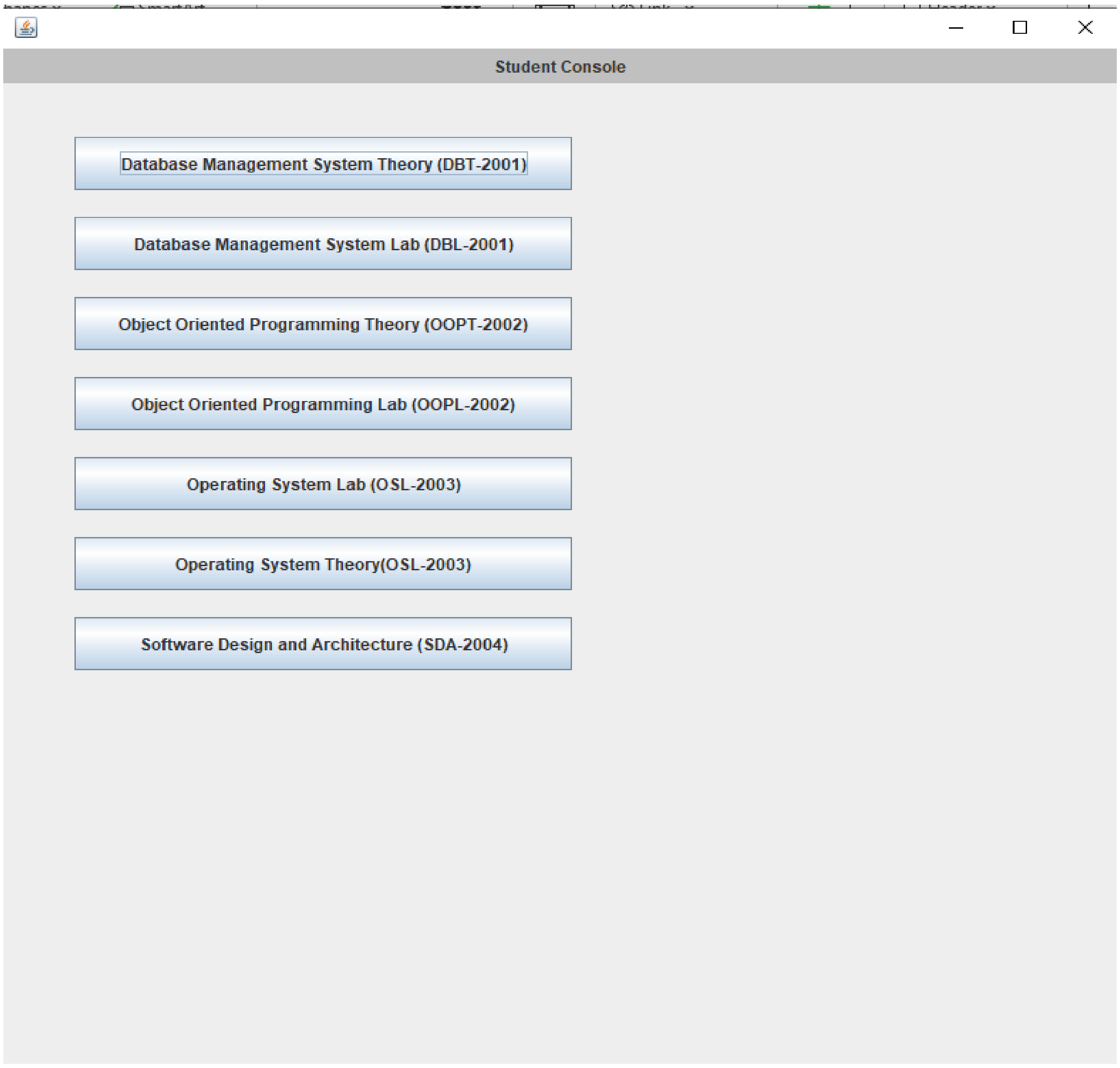


There are two button on the screen select course and view course user can click on that button and new screen will appear it will look like this


Student Course

- [ ] Database Management System Theory (DBT-2001)
- [ ] Database Management System Lab (DBL-2001)
- [ ] Object Oriented Programming Theory (OOPT-2002)
- [ ] Object Oriented Programming Lab (OOPL-2002)
- [ ] Operating System Lab (OSL-2003)
- [ ] Operating System Theory(OSL-2003)
- [ ] Software Design and Architecture (SDA-2004)

You must select at least 3 courses otherwise the system will throw an message saying you Cant enroll enroll in less then 3 courses after selescting the course the data will save in file and then the data will be read in view course


Student Console
Database Management System Theory (DBT-2001)
Database Management System Lab (DBL-2001)
Object Oriented Programming Theory (OOPT-2002)
Object Oriented Programming Lab (OOPL-2002)
Operating System Lab (OSL-2003)
Operating System Theory(OSL-2003)
Software Design and Architecture (SDA-2004)

Each button will open a new window that provides detailed information about the course. This includes the course's name, and the name of the teacher, the quiz button will only appear when the scheduled time for the quiz matches the local time of the user. Once the quiz duration has elapsed, the quiz button will disappear. This ensures that the quiz button is available for the user to access during the designated quiz period and automatically disappears once the time for the quiz has passed.

Summary

In this project, we have developed a system that manages quizzes for courses. The system includes various features and functionalities.

Firstly, we implemented a quiz creation module that allows the user to input multiple-choice questions (MCQs) along with their respective correct answers. We ensured that the system validates the input and handles error cases, such as empty questions or unanswered MCQs, by displaying appropriate error messages.

Next, we incorporated a time management component that determines the availability of the quiz. The system compares the scheduled quiz time with the local time of the user. If the times match, the quiz button becomes visible to the user, indicating that they can access the quiz. Furthermore, we implemented logic to hide the quiz button after the designated quiz duration has elapsed.

Additionally, we created a question bank where all the stored questions can be retrieved from a file. This feature allows for easy management and retrieval of questions for different courses.

Finally, we designed a course detail window that opens upon clicking the quiz button. This window provides information about the course, including the course name, the name of the teacher, and details about the quiz if it has been scheduled.

Overall, our system provides a user-friendly interface for managing quizzes, ensuring that quizzes are accessible to users during the designated timeframes and delivering relevant information about courses and their associated quizzes. I have not included all the logic screenshot I only include those which I think it was a little difficult to understand at first try the rest of the GUI logic and other logic which I used is really simple I have left the comments all over my code so miss you won’t face any problem

Important

The first screen has a problem where the GUI may not load properly. To resolve this, you can try minimizing and reopening the screen until it loads correctly. This workaround can be performed by minimizing the screen and reopening it several times until it successfully loads. This method helps to ensure that the screen loads properly and allows for a smoother user experience. The problem came on my system I suppose it will not happen the same on your system but if this happens you can follow this technique.