CME3202

CONCEPTS OF PROGRAMMING LANGUAGES

CENGONLINE

by

Gökhan Göksel Elpeze

1. Project Requirements

1.1 IDE

I used Intellij IDEA as IDE.

1.2 Classes (in OOP manner)

Classes:
-Chat
-Comment
-Course
-Database
-Message
-Post
-Stack
-Student
-Teacher
-User

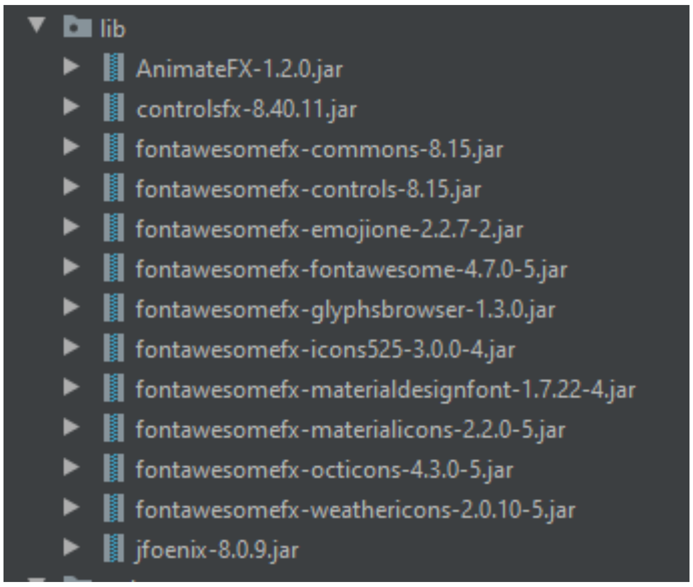
The above classes are all separate objects. There is an extend relationship between some (teacher-student <- User)

GUI controllers:
-CommentController
-CourseController
-CoursePageController
-DashboardController
-LoginController
-MainController
-MessageBoxController
-MessageController
-PostController

1.3 Libraries use used

Java.util* and

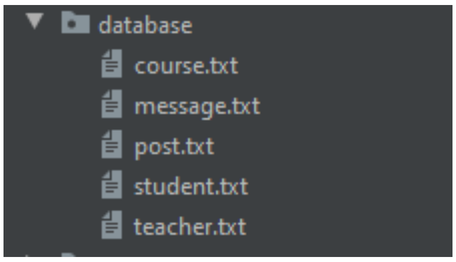
External Libraries:



```
▼ lib
    ▶ AnimateFX-1.2.0.jar
    ▶ controlsfx-8.40.11.jar
    ▶ fontawesomefx-commons-8.15.jar
    ▶ fontawesomefx-controls-8.15.jar
    ▶ fontawesomefx-emojione-2.2.7-2.jar
    ▶ fontawesomefx-fontawesome-4.7.0-5.jar
    ▶ fontawesomefx-glyphsbrowser-1.3.0.jar
    ▶ fontawesomefx-icons525-3.0.0-4.jar
    ▶ fontawesomefx-materialdesignfont-1.7.22-4.jar
    ▶ fontawesomefx-materialicons-2.2.0-5.jar
    ▶ fontawesomefx-octicons-4.3.0-5.jar
    ▶ fontawesomefx-weathericons-2.0.10-5.jar
    ▶ jfoenix-8.0.9.jar
```

1.4 Data repository

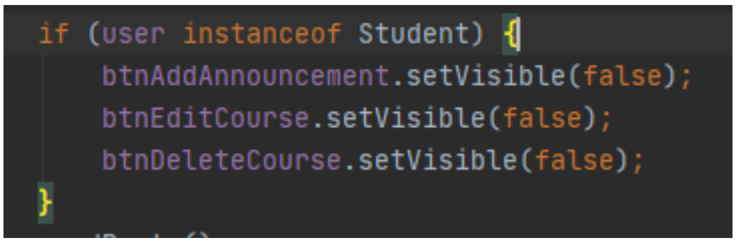
I created a class named database and I did all the reading and saving operations there. I kept the informations in txt files.



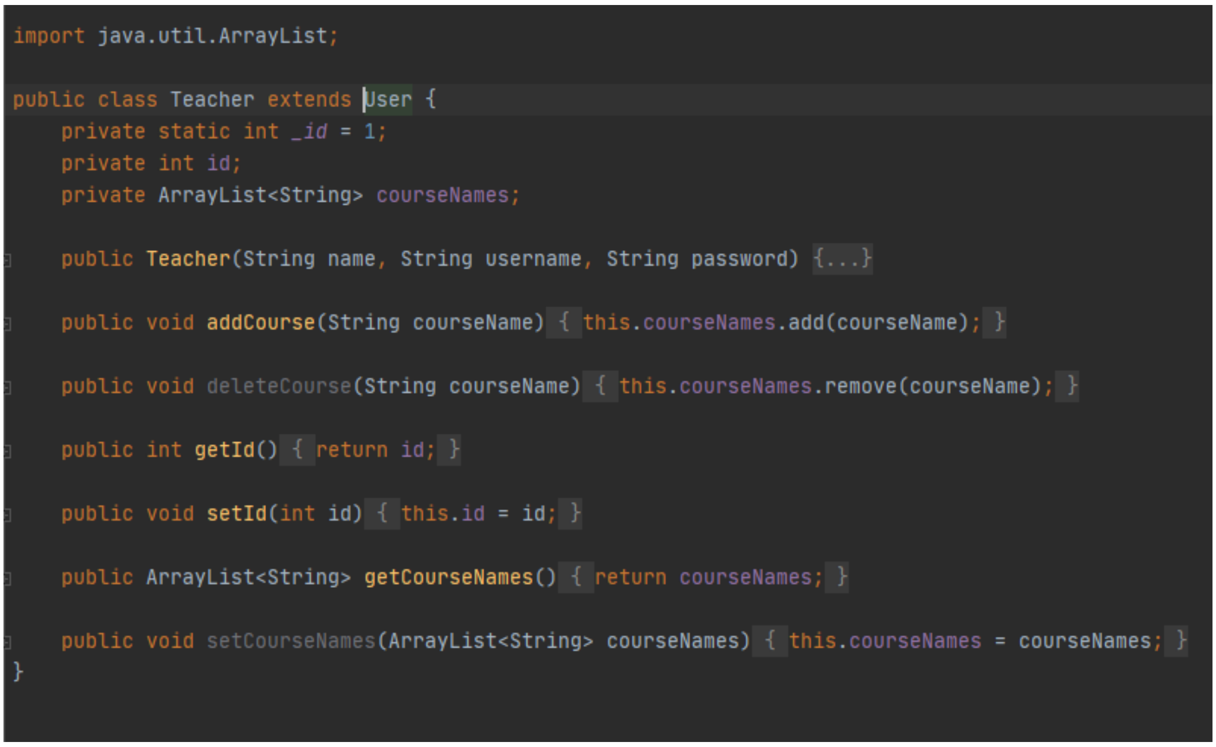
```
▼ database
      course.txt
      message.txt
      post.txt
      student.txt
      teacher.txt
```

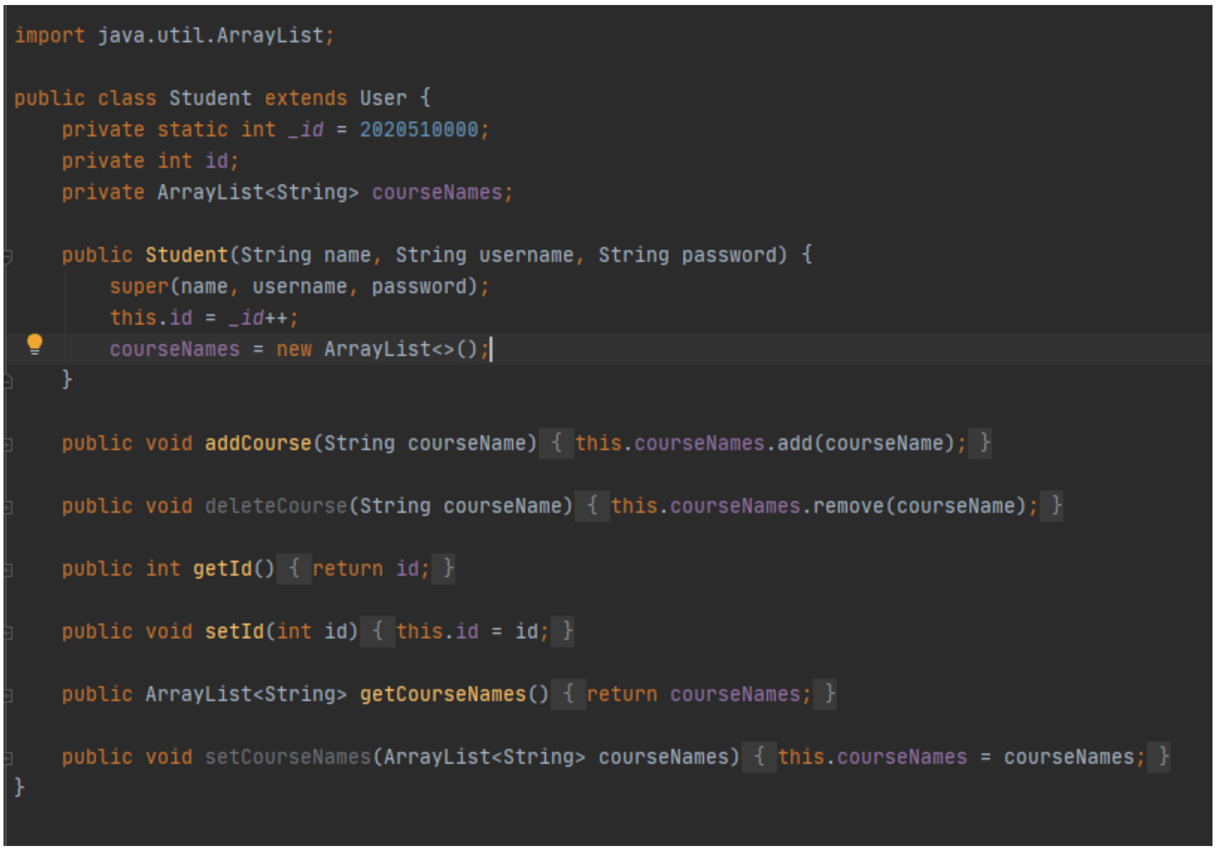
1.5 Inheritance

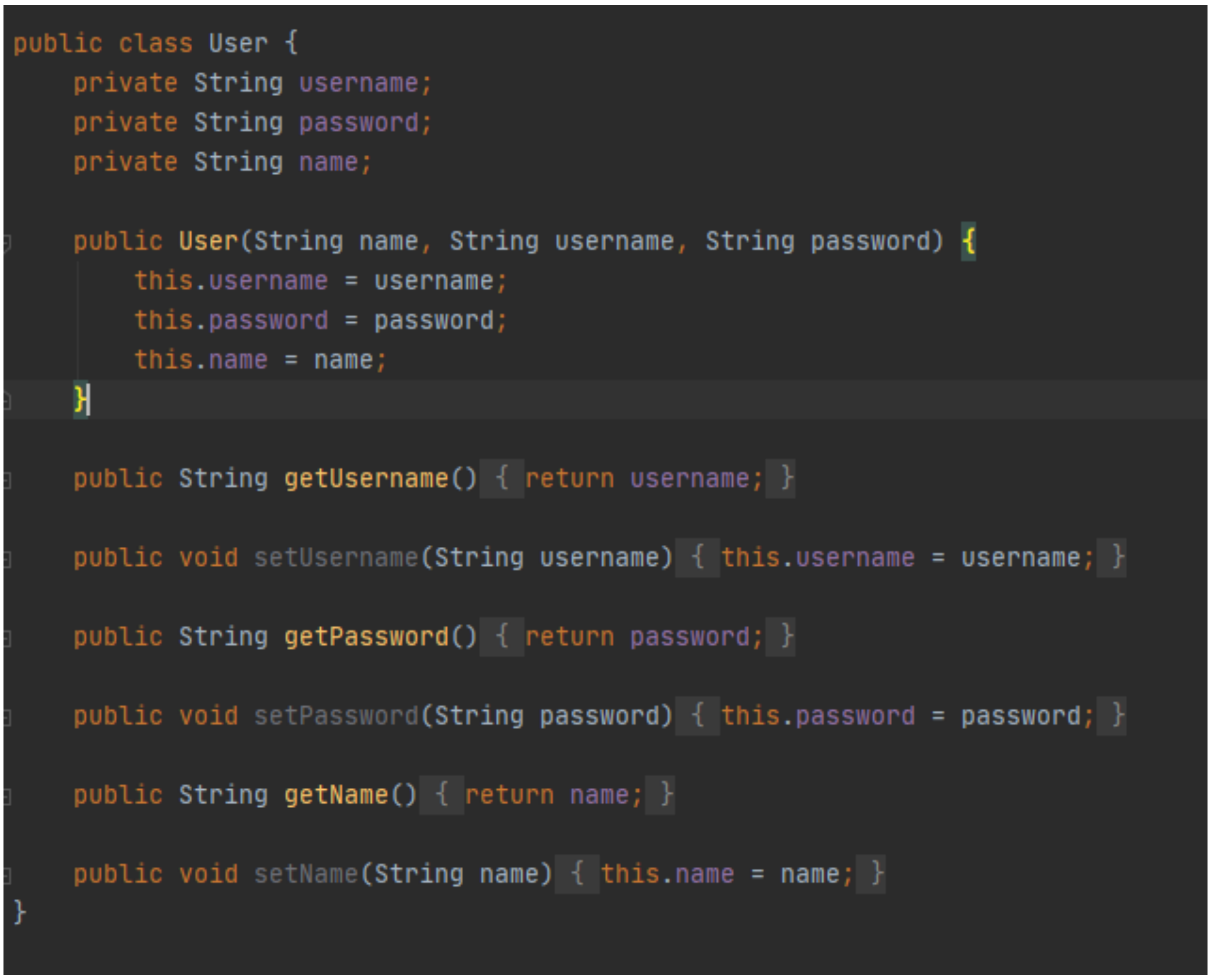
I extended the teacher and student from the user class, so I opened the buttons according to the authorization of the user who logged in like below.



```java
if (user instanceof Student) {
    btnAddAnnouncement.setVisible(false);
    btnEditCourse.setVisible(false);
    btnDeleteCourse.setVisible(false);
}
```


```java
import java.util.ArrayList;

public class Teacher extends User {
    private static int _id = 1;
    private int id;
    private ArrayList<String> courseNames;

    public Teacher(String name, String username, String password) {...}

    public void addCourse(String courseName) { this.courseNames.add(courseName); }

    public void deleteCourse(String courseName) { this.courseNames.remove(courseName); }

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public ArrayList<String> getCourseNames() { return courseNames; }

    public void setCourseNames(ArrayList<String> courseNames) { this.courseNames = courseNames; }
}
```


```java
import java.util.ArrayList;

public class Student extends User {
    private static int _id = 2020510000;
    private int id;
    private ArrayList<String> courseNames;

    public Student(String name, String username, String password) {
        super(name, username, password);
        this.id = _id++;
        courseNames = new ArrayList<>();
    }

    public void addCourse(String courseName) { this.courseNames.add(courseName); }

    public void deleteCourse(String courseName) { this.courseNames.remove(courseName); }

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public ArrayList<String> getCourseNames() { return courseNames; }

    public void setCourseNames(ArrayList<String> courseNames) { this.courseNames = courseNames; }
}
```



```
public class User {
    private String username;
    private String password;
    private String name;

    public User(String name, String username, String password) {
        this.username = username;
        this.password = password;
        this.name = name;
    }

    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }

    public String getPassword() { return password; }

    public void setPassword(String password) { this.password = password; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }
}
```

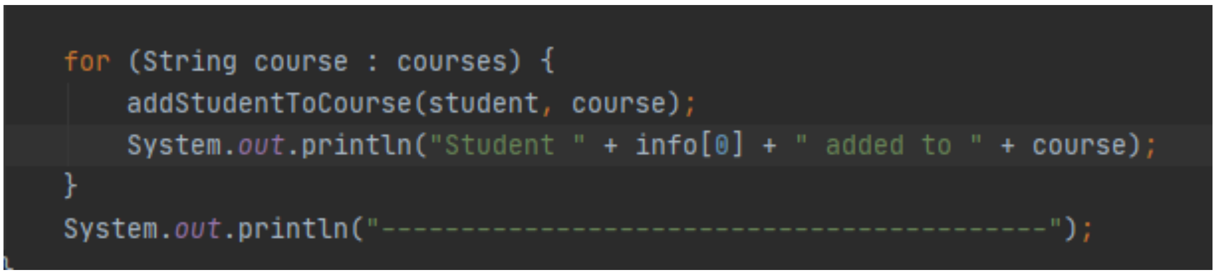
1.6 Abstract data type

I used stack for abstract data type requirement. I used for notification system. User can click notification button on GUI.

1.7 Foreach loop

I usually use foreach loop more than for loop as a habit, I think it's simpler. In this project, I used foreach loop in this Project too many times in many classes. For example:

Database class line 55 -> readStudents()



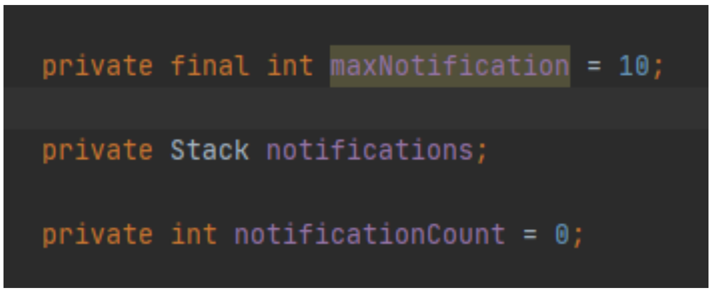
```
    for (String course : courses) {
        addStudentToCourse(student, course);
        System.out.println("Student " + info[0] + " added to " + course);
    }
    System.out.println("--------------------------------------------");
```

1.8 Switch-case condition

I couldn't find a place to use. I did not need

1.9 Named constants

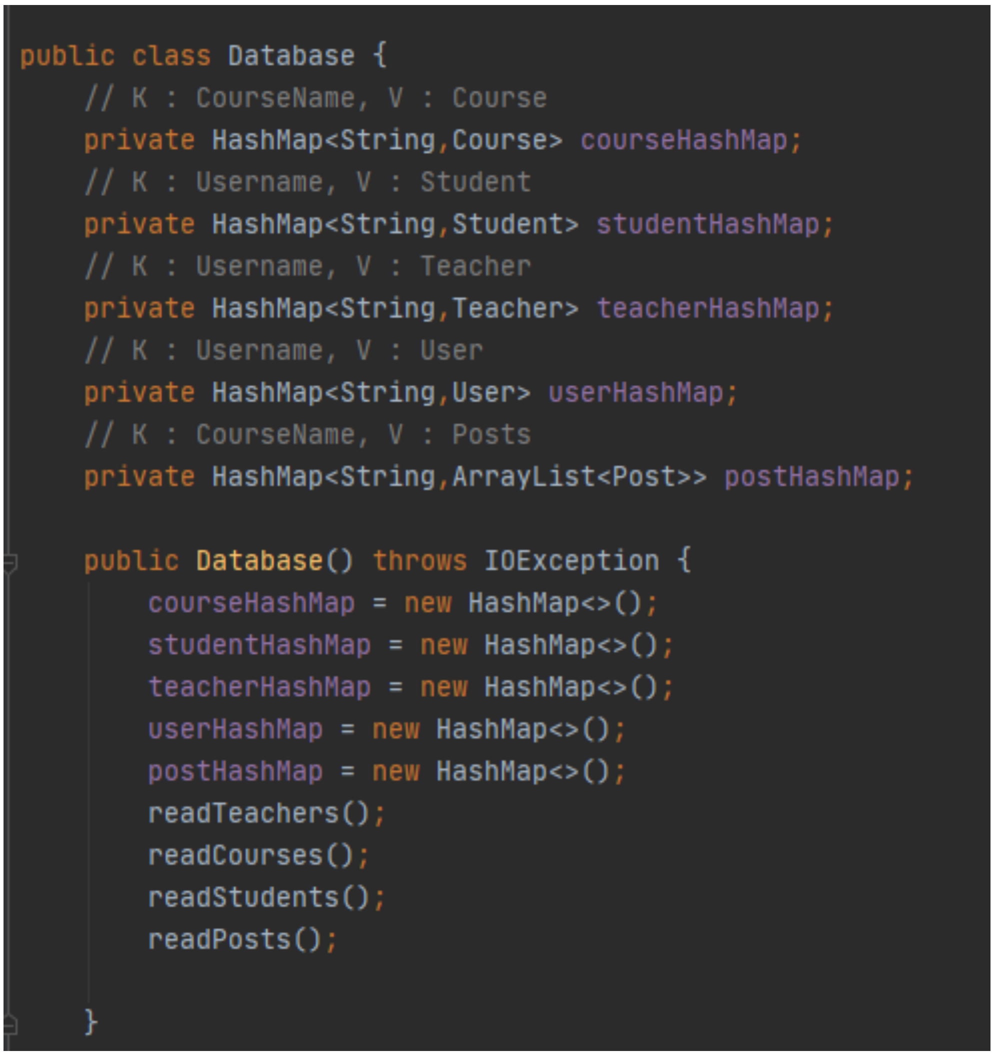
I restricted the maximum number of notifications that can come.



```
private final int maxNotification = 10;

private Stack notifications;

private int notificationCount = 0;
```

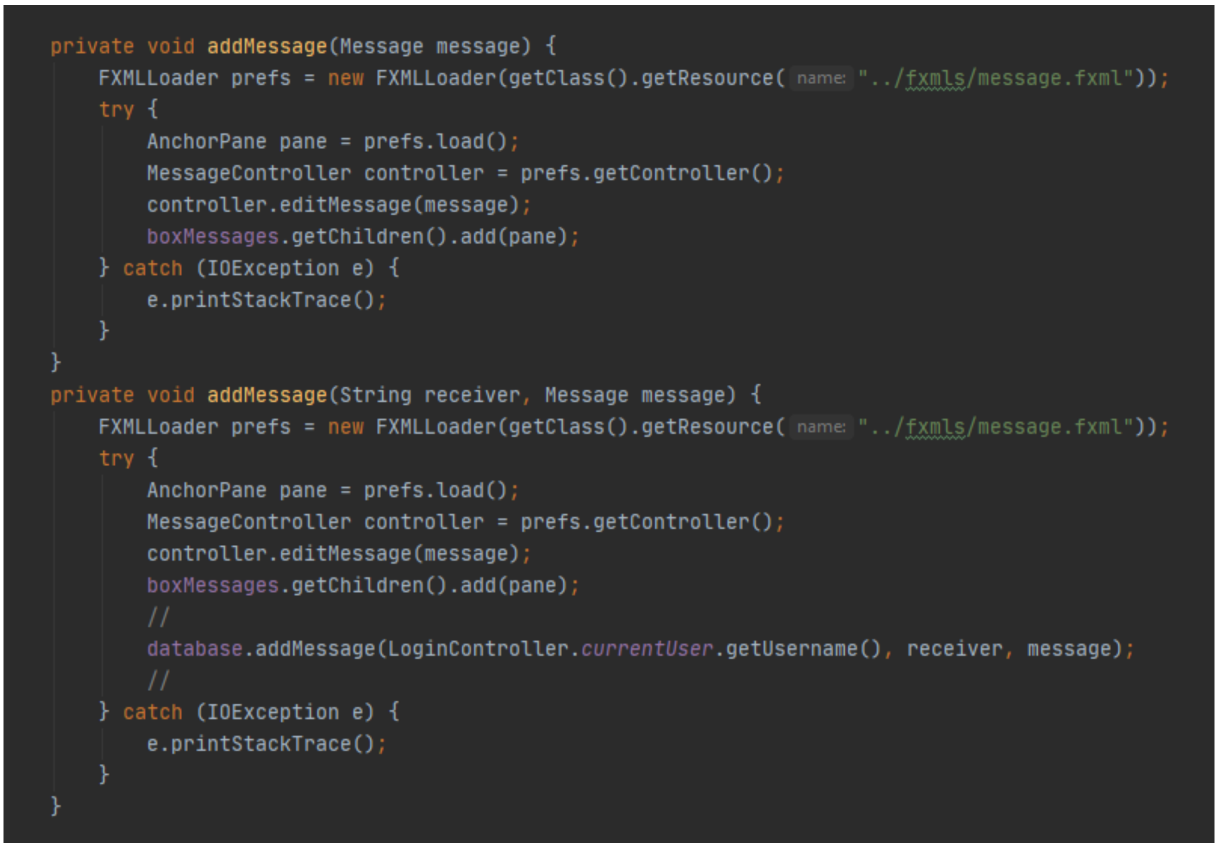
1.10 Associative Arrays

As you know Java doesn't support associative arrays, however this could easily be achieved using a Map. Its used to implement an associative array, a structure that can map keys to values. So i used Hashmap's to implement this.



```
public class Database {
    // K : CourseName, V : Course
    private HashMap<String,Course> courseHashMap;
    // K : Username, V : Student
    private HashMap<String,Student> studentHashMap;
    // K : Username, V : Teacher
    private HashMap<String,Teacher> teacherHashMap;
    // K : Username, V : User
    private HashMap<String,User> userHashMap;
    // K : CourseName, V : Posts
    private HashMap<String,ArrayList<Post>> postHashMap;

    public Database() throws IOException {
        courseHashMap = new HashMap<>();
        studentHashMap = new HashMap<>();
        teacherHashMap = new HashMap<>();
        userHashMap = new HashMap<>();
        postHashMap = new HashMap<>();
        readTeachers();
        readCourses();
        readStudents();
        readPosts();

    }
```

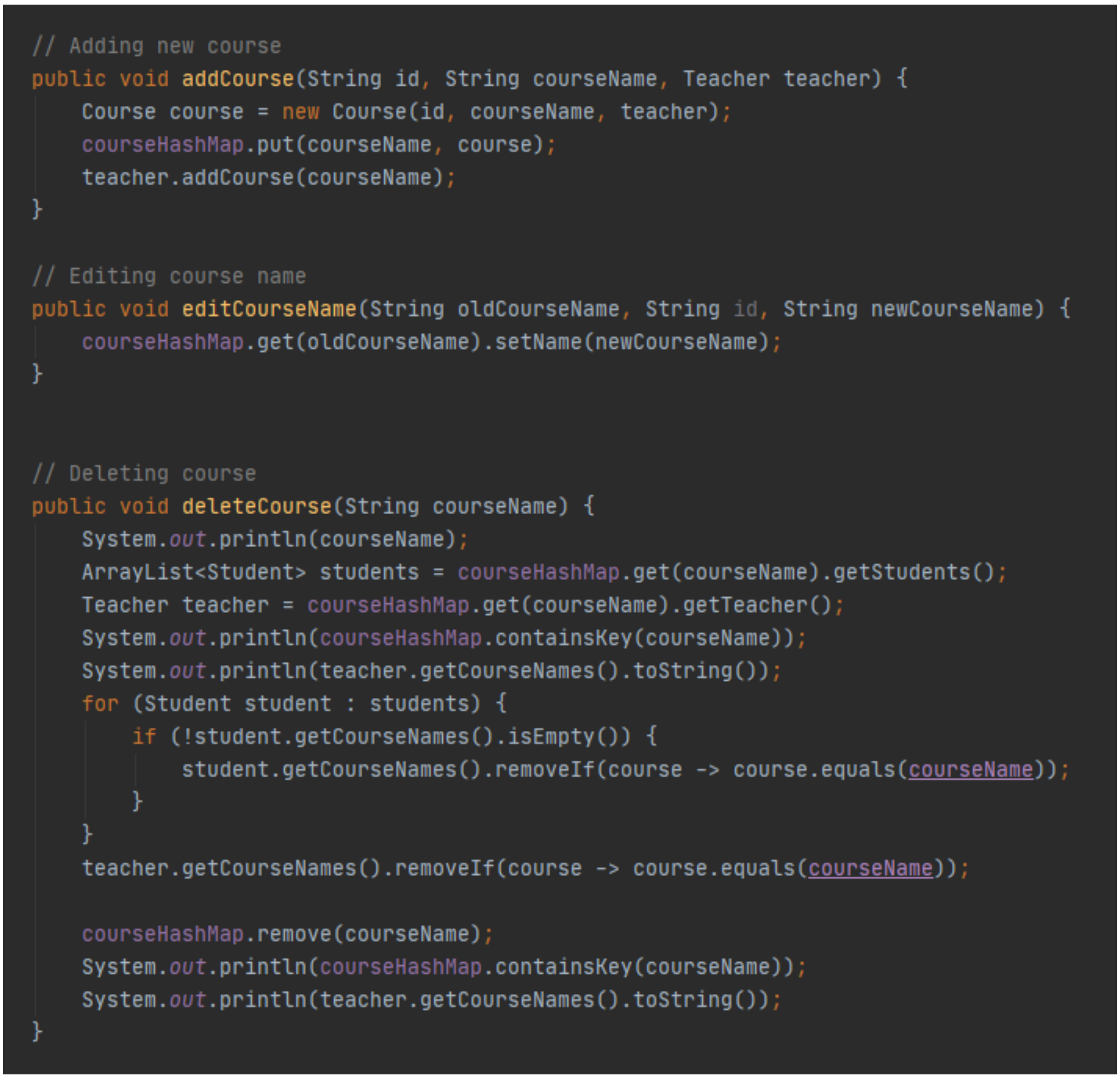
1.11 Method Overloading



```java
private void addMessage(Message message) {
    FXMLLoader prefs = new FXMLLoader(getClass().getResource( name: "../fxmls/message.fxml"));
    try {
        AnchorPane pane = prefs.load();
        MessageController controller = prefs.getController();
        controller.editMessage(message);
        boxMessages.getChildren().add(pane);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
private void addMessage(String receiver, Message message) {
    FXMLLoader prefs = new FXMLLoader(getClass().getResource( name: "../fxmls/message.fxml"));
    try {
        AnchorPane pane = prefs.load();
        MessageController controller = prefs.getController();
        controller.editMessage(message);
        boxMessages.getChildren().add(pane);
        //
        database.addMessage(LoginController.currentUser.getUsername(), receiver, message);
        //
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

2. UI / Login as a Teacher

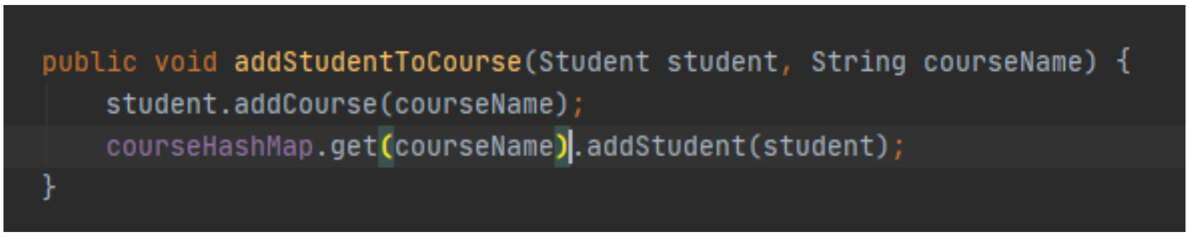
2.1 Add/edit/delete a course



```java
// Adding new course
public void addCourse(String id, String courseName, Teacher teacher) {
    Course course = new Course(id, courseName, teacher);
    courseHashMap.put(courseName, course);
    teacher.addCourse(courseName);
}

// Editing course name
public void editCourseName(String oldCourseName, String id, String newCourseName) {
    courseHashMap.get(oldCourseName).setName(newCourseName);
}


// Deleting course
public void deleteCourse(String courseName) {
    System.out.println(courseName);
    ArrayList<Student> students = courseHashMap.get(courseName).getStudents();
    Teacher teacher = courseHashMap.get(courseName).getTeacher();
    System.out.println(courseHashMap.containsKey(courseName));
    System.out.println(teacher.getCourseNames().toString());
    for (Student student : students) {
        if (!student.getCourseNames().isEmpty()) {
            student.getCourseNames().removeIf(course -> course.equals(courseName));
        }
    }
    teacher.getCourseNames().removeIf(course -> course.equals(courseName));

    courseHashMap.remove(courseName);
    System.out.println(courseHashMap.containsKey(courseName));
    System.out.println(teacher.getCourseNames().toString());
}
```

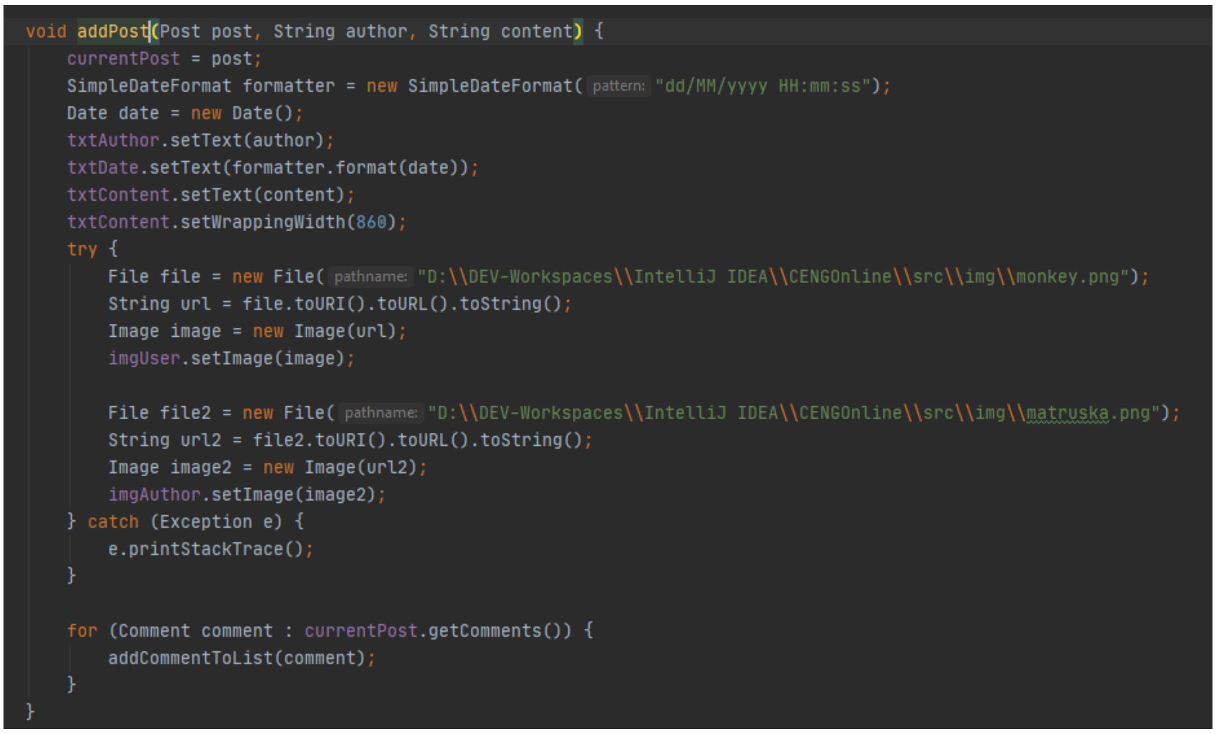
2.2 Enroll students (not necessary!)

Teacher can add student to course.

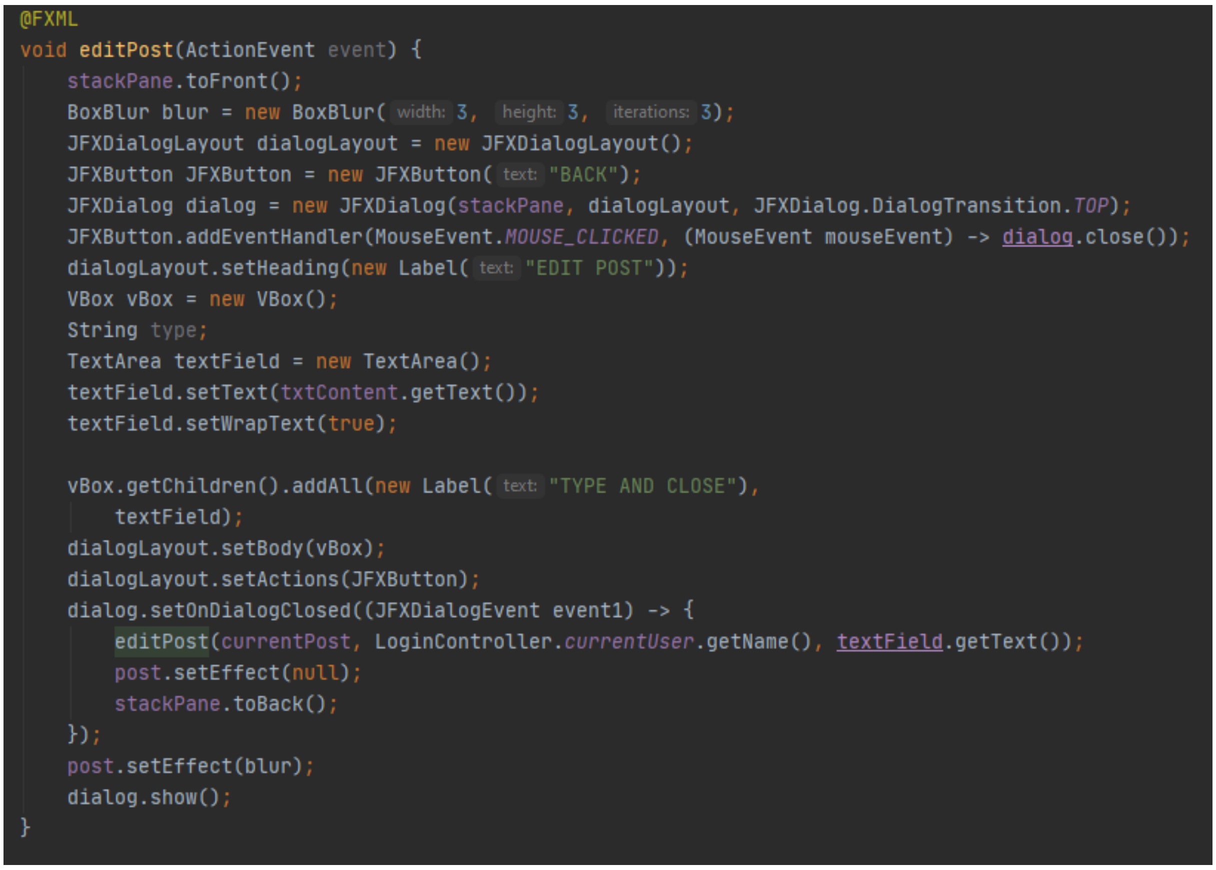


```java
public void addStudentToCourse(Student student, String courseName) {
    student.addCourse(courseName);
    courseHashMap.get(courseName).addStudent(student);
}
```

2.3 Add/edit/delete an assignment



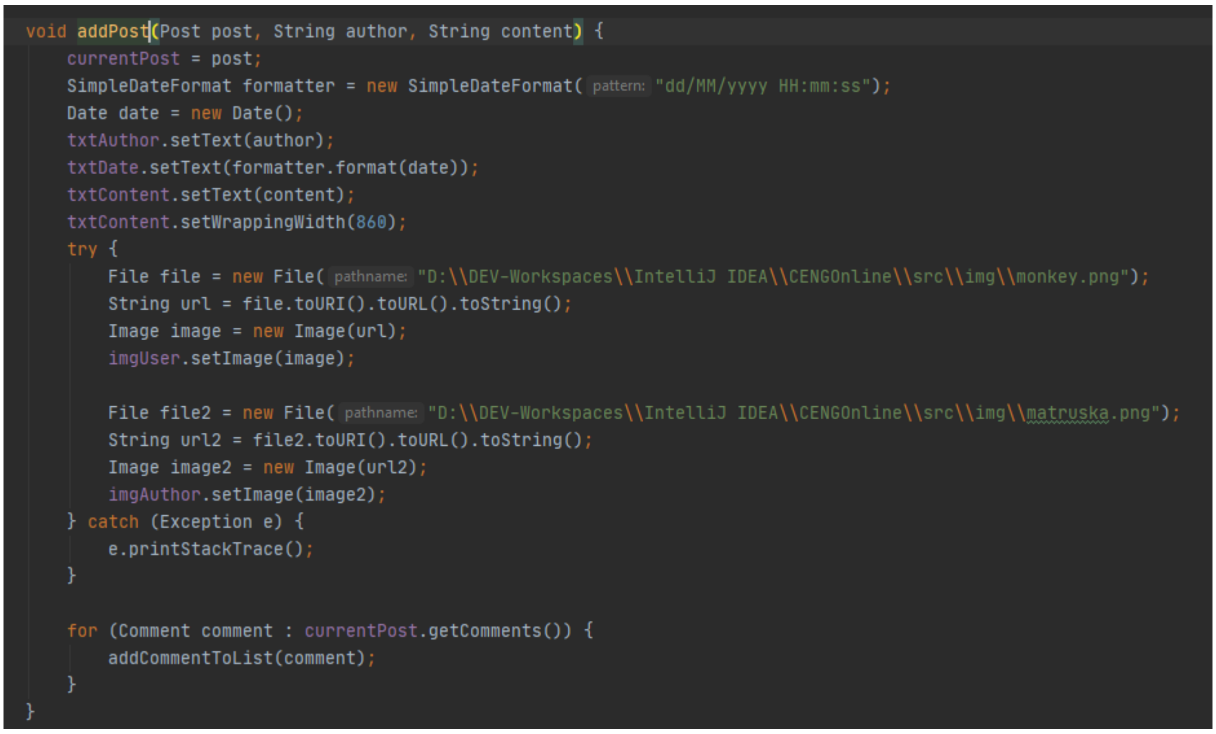
```java
void addPost(Post post, String author, String content) {
    currentPost = post;
    SimpleDateFormat formatter = new SimpleDateFormat( pattern: "dd/MM/yyyy HH:mm:ss");
    Date date = new Date();
    txtAuthor.setText(author);
    txtDate.setText(formatter.format(date));
    txtContent.setText(content);
    txtContent.setWrappingWidth(860);
    try {
        File file = new File( pathname: "D:\\DEV-Workspaces\\IntelliJ IDEA\\CENGOnline\\src\\img\\monkey.png");
        String url = file.toURI().toURL().toString();
        Image image = new Image(url);
        imgUser.setImage(image);

        File file2 = new File( pathname: "D:\\DEV-Workspaces\\IntelliJ IDEA\\CENGOnline\\src\\img\\matruska.png");
        String url2 = file2.toURI().toURL().toString();
        Image image2 = new Image(url2);
        imgAuthor.setImage(image2);
    } catch (Exception e) {
        e.printStackTrace();
    }

    for (Comment comment : currentPost.getComments()) {
        addCommentToList(comment);
    }
}
```



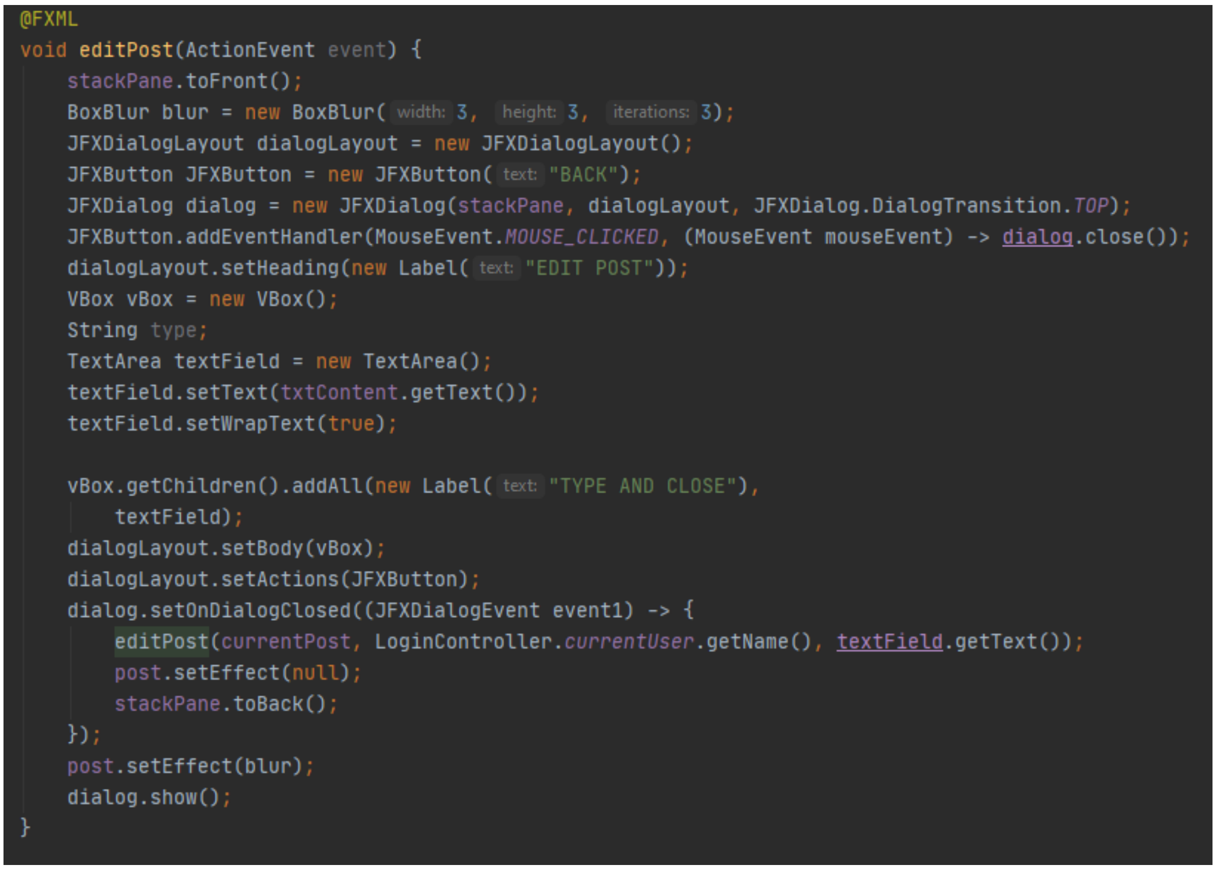
```java
@FXML
void editPost(ActionEvent event) {
    stackPane.toFront();
    BoxBlur blur = new BoxBlur( width: 3,  height: 3,  iterations: 3);
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
    JFXButton JFXButton = new JFXButton( text: "BACK");
    JFXDialog dialog = new JFXDialog(stackPane, dialogLayout, JFXDialog.DialogTransition.TOP);
    JFXButton.addEventHandler(MouseEvent.MOUSE_CLICKED, (MouseEvent mouseEvent) -> dialog.close());
    dialogLayout.setHeading(new Label( text: "EDIT POST"));
    VBox vBox = new VBox();
    String type;
    TextArea textField = new TextArea();
    textField.setText(txtContent.getText());
    textField.setWrapText(true);

    vBox.getChildren().addAll(new Label( text: "TYPE AND CLOSE"),
        textField);
    dialogLayout.setBody(vBox);
    dialogLayout.setActions(JFXButton);
    dialog.setOnDialogClosed((JFXDialogEvent event1) -> {
        editPost(currentPost, LoginController.currentUser.getName(), textField.getText());
        post.setEffect(null);
        stackPane.toBack();
    });
    post.setEffect(blur);
    dialog.show();
}
```

2.4 View submitted works by students

2.5 Add/edit/delete an announcement



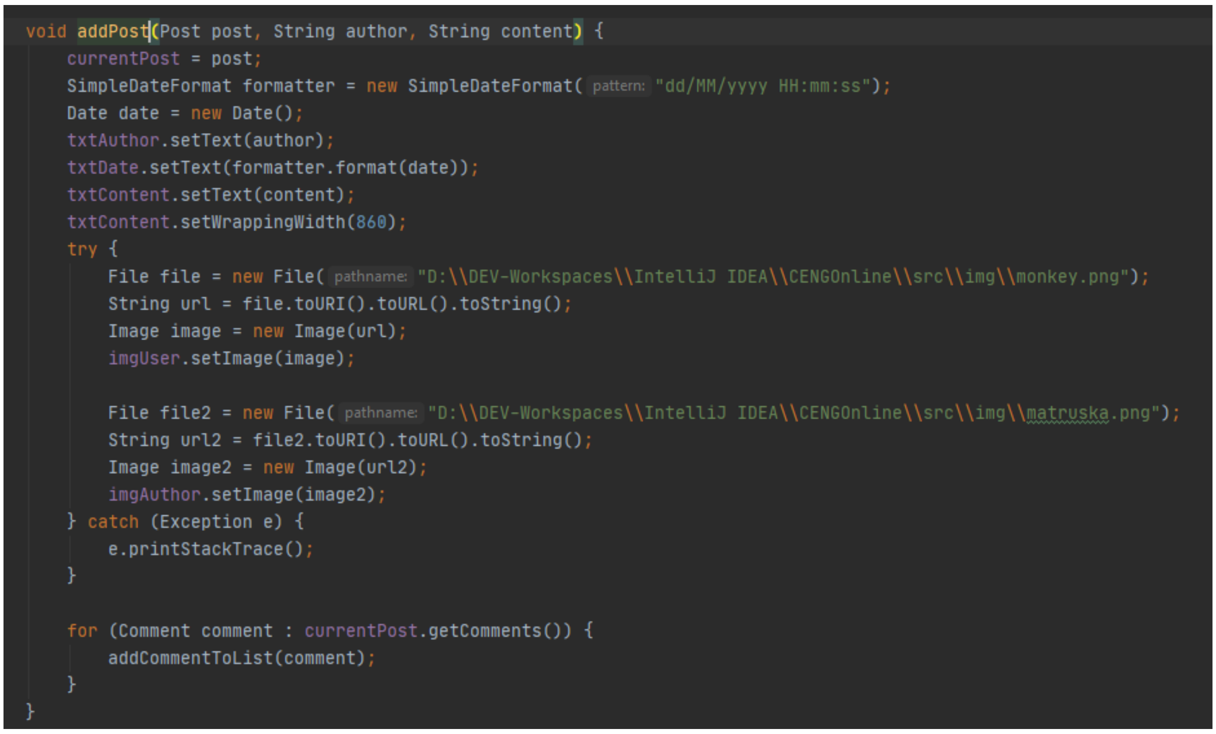
```
void addPost(Post post, String author, String content) {
    currentPost = post;
    SimpleDateFormat formatter = new SimpleDateFormat( pattern: "dd/MM/yyyy HH:mm:ss");
    Date date = new Date();
    txtAuthor.setText(author);
    txtDate.setText(formatter.format(date));
    txtContent.setText(content);
    txtContent.setWrappingWidth(860);
    try {
        File file = new File( pathname: "D:\\DEV-Workspaces\\IntelliJ IDEA\\CENGOnline\\src\\img\\monkey.png");
        String url = file.toURI().toURL().toString();
        Image image = new Image(url);
        imgUser.setImage(image);

        File file2 = new File( pathname: "D:\\DEV-Workspaces\\IntelliJ IDEA\\CENGOnline\\src\\img\\matruska.png");
        String url2 = file2.toURI().toURL().toString();
        Image image2 = new Image(url2);
        imgAuthor.setImage(image2);
    } catch (Exception e) {
        e.printStackTrace();
    }

    for (Comment comment : currentPost.getComments()) {
        addCommentToList(comment);
    }
}
```



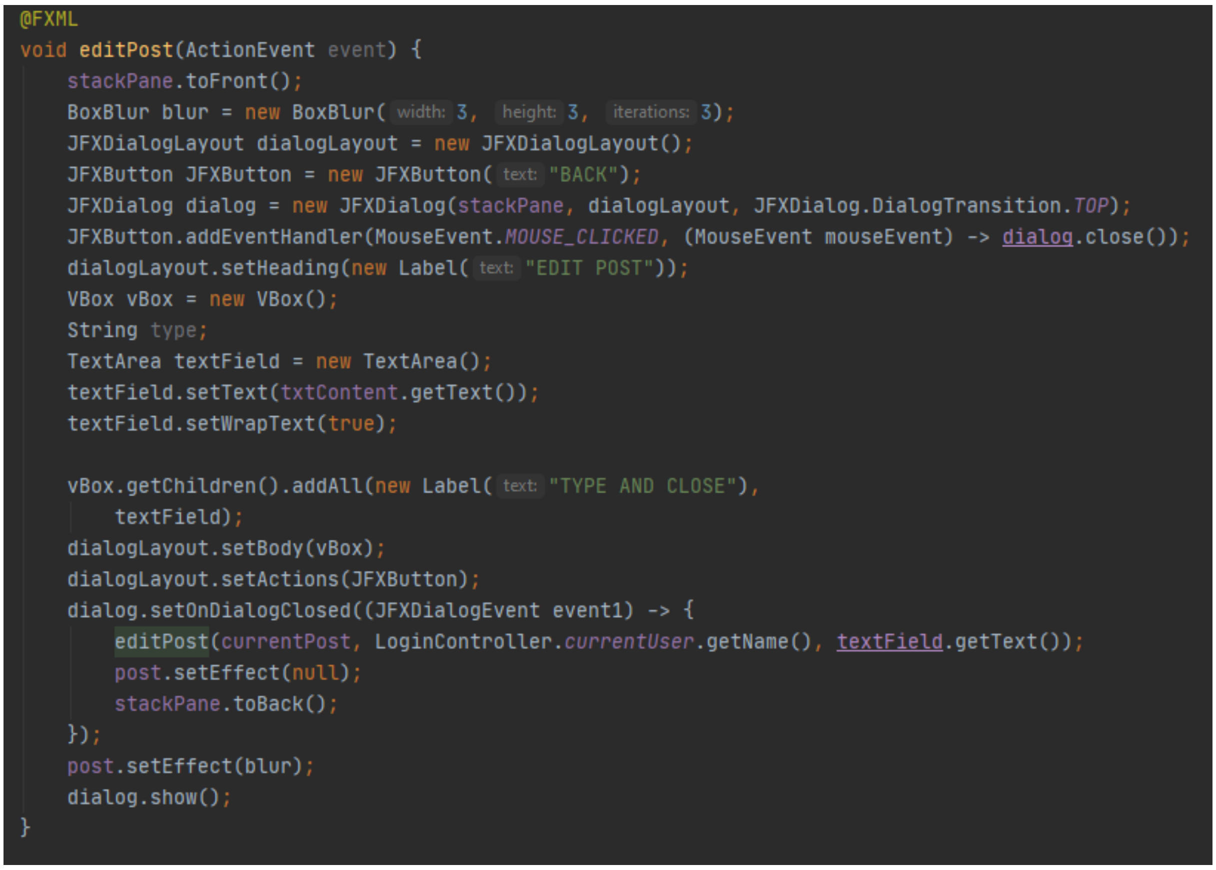
```
@FXML
void editPost(ActionEvent event) {
    stackPane.toFront();
    BoxBlur blur = new BoxBlur( width: 3,  height: 3,  iterations: 3);
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
    JFXButton JFXButton = new JFXButton( text: "BACK");
    JFXDialog dialog = new JFXDialog(stackPane, dialogLayout, JFXDialog.DialogTransition.TOP);
    JFXButton.addEventHandler(MouseEvent.MOUSE_CLICKED, (MouseEvent mouseEvent) -> dialog.close());
    dialogLayout.setHeading(new Label( text: "EDIT POST"));
    VBox vBox = new VBox();
    String type;
    TextArea textField = new TextArea();
    textField.setText(txtContent.getText());
    textField.setWrapText(true);

    vBox.getChildren().addAll(new Label( text: "TYPE AND CLOSE"),
        textField);
    dialogLayout.setBody(vBox);
    dialogLayout.setActions(JFXButton);
    dialog.setOnDialogClosed((JFXDialogEvent event1) -> {
        editPost(currentPost, LoginController.currentUser.getName(), textField.getText());
        post.setEffect(null);
        stackPane.toBack();
    });
    post.setEffect(blur);
    dialog.show();
}
```

2.6 Send a message and view incoming messages

2.7 Add/edit/delete a post.

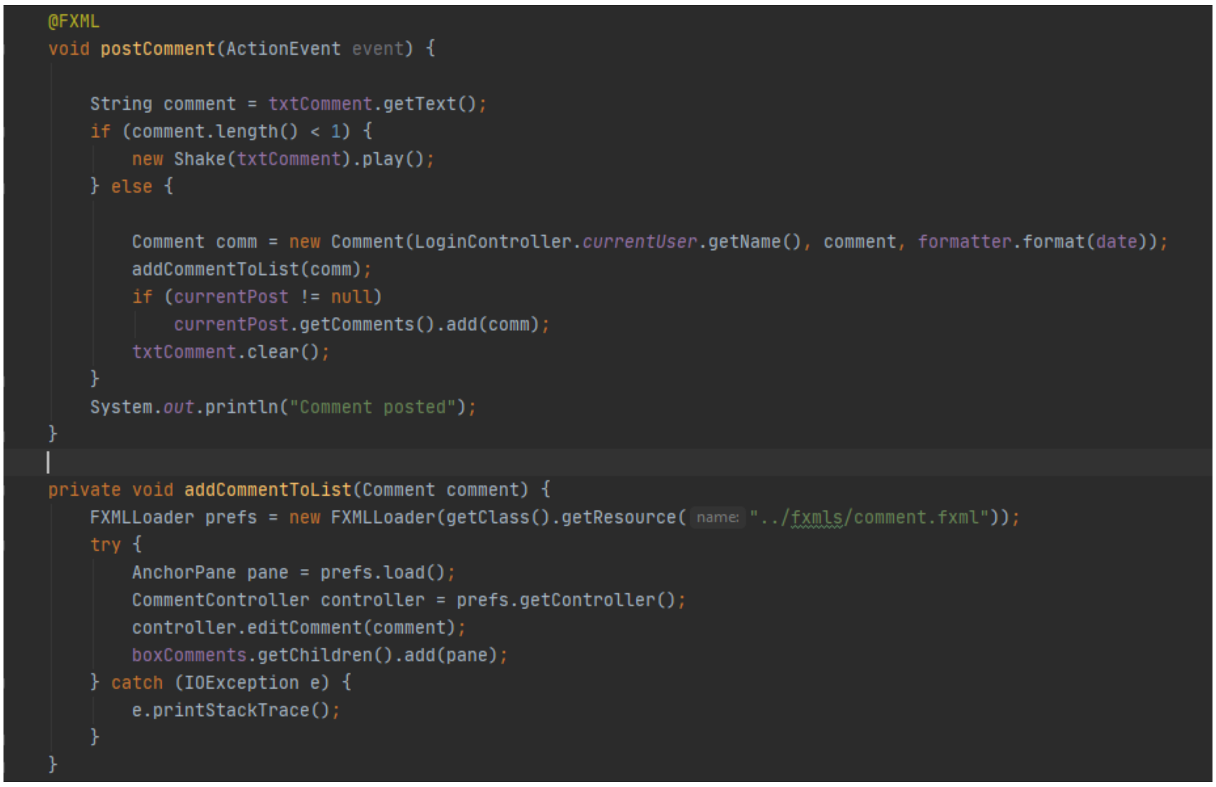


```java
void addPost(Post post, String author, String content) {
    currentPost = post;
    SimpleDateFormat formatter = new SimpleDateFormat( pattern: "dd/MM/yyyy HH:mm:ss");
    Date date = new Date();
    txtAuthor.setText(author);
    txtDate.setText(formatter.format(date));
    txtContent.setText(content);
    txtContent.setWrappingWidth(860);
    try {
        File file = new File( pathname: "D:\\DEV-Workspaces\\IntelliJ IDEA\\CENGOnline\\src\\img\\monkey.png");
        String url = file.toURI().toURL().toString();
        Image image = new Image(url);
        imgUser.setImage(image);

        File file2 = new File( pathname: "D:\\DEV-Workspaces\\IntelliJ IDEA\\CENGOnline\\src\\img\\matruska.png");
        String url2 = file2.toURI().toURL().toString();
        Image image2 = new Image(url2);
        imgAuthor.setImage(image2);
    } catch (Exception e) {
        e.printStackTrace();
    }

    for (Comment comment : currentPost.getComments()) {
        addCommentToList(comment);
    }
}
```



```java
@FXML
void editPost(ActionEvent event) {
    stackPane.toFront();
    BoxBlur blur = new BoxBlur( width: 3,  height: 3,  iterations: 3);
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
    JFXButton JFXButton = new JFXButton( text: "BACK");
    JFXDialog dialog = new JFXDialog(stackPane, dialogLayout, JFXDialog.DialogTransition.TOP);
    JFXButton.addEventHandler(MouseEvent.MOUSE_CLICKED, (MouseEvent mouseEvent) -> dialog.close());
    dialogLayout.setHeading(new Label( text: "EDIT POST"));
    VBox vBox = new VBox();
    String type;
    TextArea textField = new TextArea();
    textField.setText(txtContent.getText());
    textField.setWrapText(true);

    vBox.getChildren().addAll(new Label( text: "TYPE AND CLOSE"),
        textField);
    dialogLayout.setBody(vBox);
    dialogLayout.setActions(JFXButton);
    dialog.setOnDialogClosed((JFXDialogEvent event1) -> {
        editPost(currentPost, LoginController.currentUser.getName(), textField.getText());
        post.setEffect(null);
        stackPane.toBack();
    });
    post.setEffect(blur);
    dialog.show();
}
```

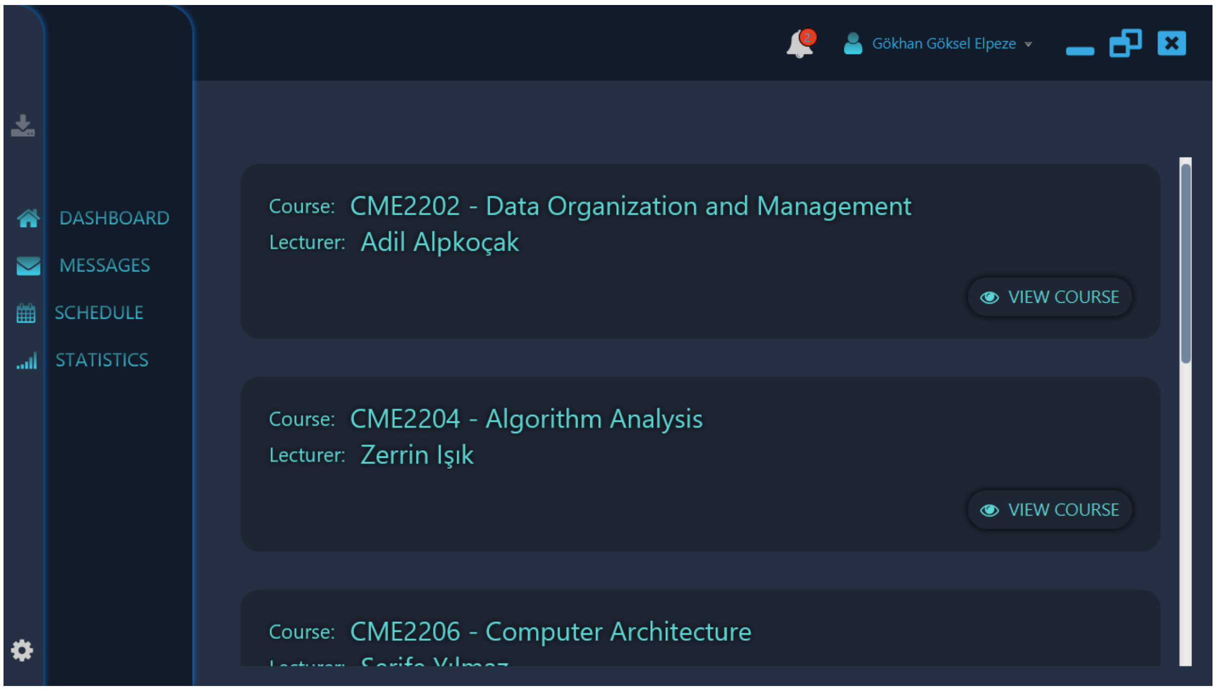
2.8 Comment to a post.



```
@FXML
void postComment(ActionEvent event) {

    String comment = txtComment.getText();
    if (comment.length() < 1) {
        new Shake(txtComment).play();
    } else {

        Comment comm = new Comment(LoginController.currentUser.getName(), comment, formatter.format(date));
        addCommentToList(comm);
        if (currentPost != null)
            currentPost.getComments().add(comm);
        txtComment.clear();
    }
    System.out.println("Comment posted");
}

private void addCommentToList(Comment comment) {
    FXMLLoader prefs = new FXMLLoader(getClass().getResource( name: "../fxmls/comment.fxml"));
    try {
        AnchorPane pane = prefs.load();
        CommentController controller = prefs.getController();
        controller.editComment(comment);
        boxComments.getChildren().add(pane);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

3. UI / Login as a Student

3.1 View the list of enrolled courses.



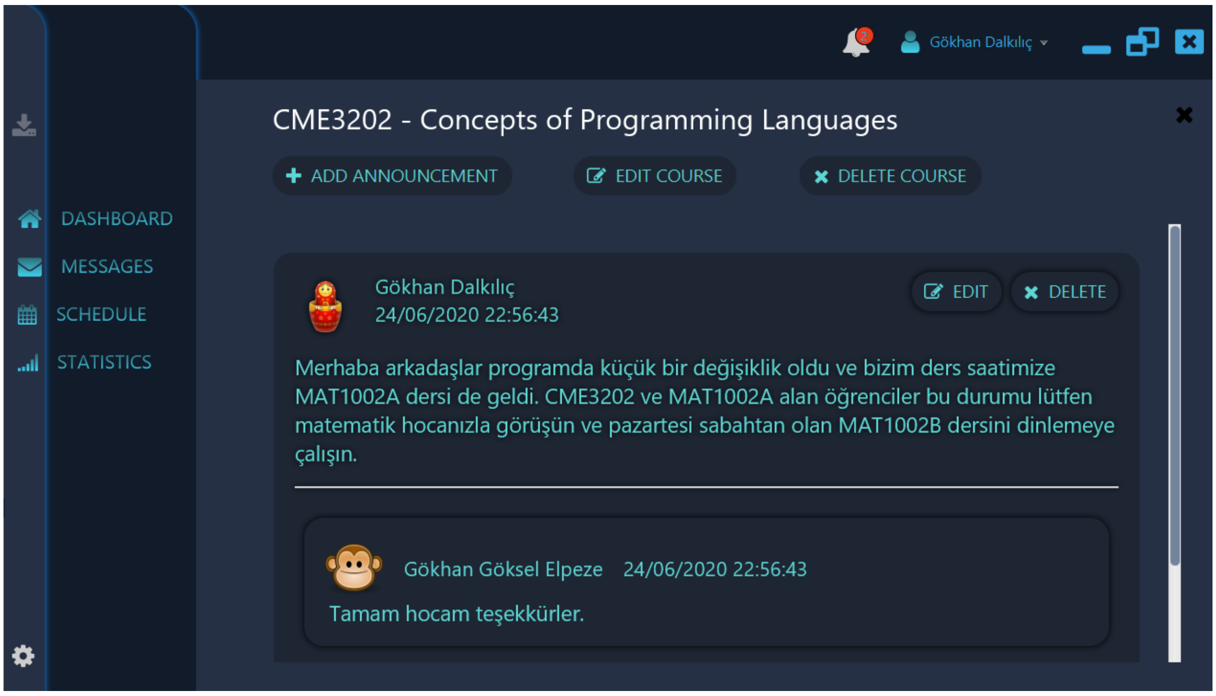
Course: CME2202 - Data Organization and Management
Lecturer: Adil Alpkoçak

Course: CME2204 - Algorithm Analysis
Lecturer: Zerrin Işık

Course: CME2206 - Computer Architecture

3.2 View the assignments and upload a work

3.3 View the announcements



CME3202 - Concepts of Programming Languages

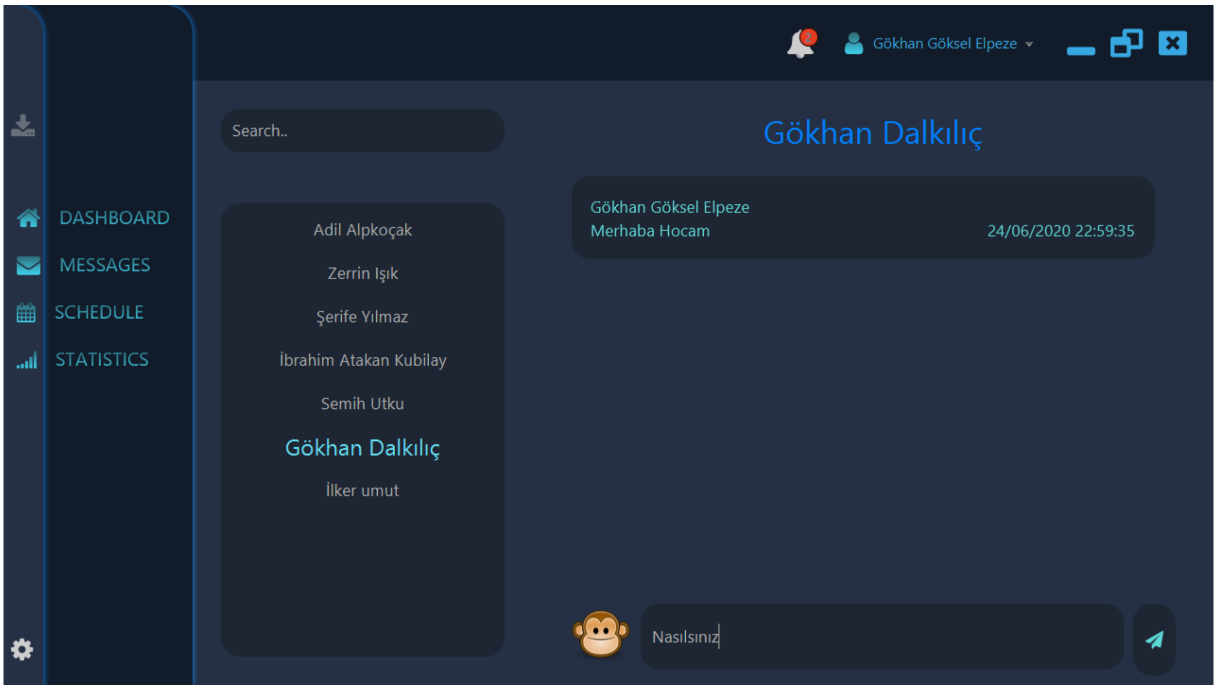
Gökhan Dalkılıç
24/06/2020 22:56:43

Merhaba arkadaşlar programda küçük bir değişiklik oldu ve bizim ders saatimize MAT1002A dersi de geldi. CME3202 ve MAT1002A alan öğrenciler bu durumu lütfen matematik hocanızla görüşün ve pazartesi sabahtan olan MAT1002B dersini dinlemeye çalışın.

Gökhan Göksel Elpeze 24/06/2020 22:56:43

Tamam hocam teşekkürler.

3.4 Send a message and view incoming messages

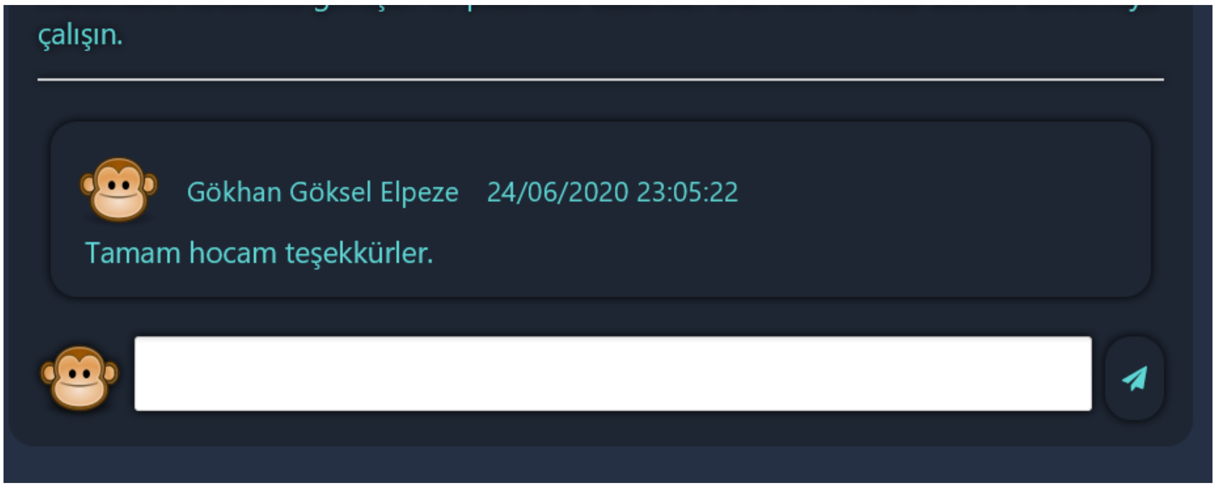


Gökhan Dalkılıç

Gökhan Göksel Elpeze
Merhaba Hocam 24/06/2020 22:59:35

Nasılsınız

3.5 Comment to a post.


çalışın.

Gökhan Göksel Elpeze 24/06/2020 23:05:22

Tamam hocam teşekkürler.

4. REMAININGS

4.1 Contribution of each group member

I did the homework alone, I don't have a group friend.

4.2 Incomplete tasks: reasons, explanations

Assignment part

Database part cant work properly.

4.3 Additional improvements

Notification System, Animations

4.4 Problems encountered

Database management