

# ws45CodeReview

martes, 7 de marzo de 2023 8:34

**Team 1:** Scriptal

**Github:** <https://github.com/KarlaAns/Scriptal-Repository>

**Project name:** Student System

**Leader:** Karla Ansatuña

**MongoDB URI:**

mongodb+srv://laandrade:laandrade@cluster0.jcz1lsa.mongodb.net/test

**Inspector:** T10 Justin Villarroel

- 1 ACALO CRUZ CRISTIAN JOSE
- 2 ALDAZ VIDAL BERNARDO ANDRE
- 3 ANDRADE ENCALADA LUIS ALEJANDRO
- 4 ANSATUÑA ANDRADE KARLA ALEJANDRA

Code Quality -20/-20 every pitfall -1  
-20+15 = -5/-20

Committed code

```
private void txtNameActionPerformed(Javax.swing.event.ActionEvent e) {  
    // TODO add your handling code here:  
    //Document student = findDataStudent();  
    txtName.setText();  
    dtShipping.setDate(date);  
    dtDeadline.setDate(date);  
}  
  
private void cmbTypeActionPerformed(Javax.swing.event.ActionEvent e) {  
    // TODO add your handling code here:  
}  
  
159 private void txtNameKeyPressed(Javax.swing.event.KeyEvent e) {  
160     // TODO add your handling code here:  
161     //char validate = e.getKeyChar();  
162     //Character.isLetter(validate);  
163     getFocus().beep();  
164     e.consume();  
165     JOptionPane.showMessageDialog(this, "The inserted class already exist", "Classroom insertion", JOptionPane.WARNING_MESSAGE);  
166 }  
167  
168  
private void btnInClassroomsActionPerformed(Javax.swing.event.ActionEvent e) {  
    // TODO add your handling code here:  
    //Document teacherId = findTeacherById(teacher.getId());  
    FrmClassroomManagement classroomManagement = new FrmClassroomManagement(teacherId);  
    classroomManagement.setVisible(true);  
    if ("FlatLight".equals(e.getSource().getClass().getName())) {  
        classroomManagement.setStatusCbnDarkMode(selected false);  
    } else {  
        classroomManagement.setStatusCbnDarkMode(selected true);  
    }  
}
```

```

        classroomManagement.setStatusCbnDarkNodeC.isSelected trueX;
    }
    this.dispose();
}

private void btnEnrollmentActionPerformed(java.awt.event.ActionEvent evt) {
    FrmEnrollment frmEnrollment = new FrmEnrollment();
    frmEnrollment.setVisible = trueX;
    System.out.println(" UIManager.getLookAndFeel().getName()");
    if ("FlatLafLight".equalsC( UIManager.getLookAndFeel().getName())) {
        frmEnrollment.setStatusCbnDarkNodeC.isSelected falseX;
    } else {
        //frmEnrollment.setStatusCbnDarkNodeC.isSelected trueX;
    }
    this.dispose();
}

```

event controller

```

public static void deleteEventOntIdC {
    //int idDigit = Integer.parseInt(id);
    MongoClient collection = MongoClient.getConnectionC(selected "events"X);
    Bson filter = Filters.eqC("id", new String.valueOf(id));
    collection.deleteOneC(filter);
}

```

duplicate code

```

private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    FrmLogin login = new FrmLogin();
    login.setVisible = trueX;
    if ("FlatLafLight".equalsC( UIManager.getLookAndFeel().getName())) {
        login.setStatusCbnDarkNodeC.isSelected falseX;
    } else {
        login.setStatusCbnDarkNodeC.isSelected trueX;
    }
    this.dispose();
}

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    FrmCreateEvent btnAdd = new FrmCreateEvent();
    btnAdd.setVisible = trueX;
    if ("FlatLafLight".equalsC( UIManager.getLookAndFeel().getName())) {
        btnAdd.setStatusCbnDarkNodeC.isSelected falseX;
    } else {
        btnAdd.setStatusCbnDarkNodeC.isSelected trueX;
    }
    this.dispose();
}

private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    //
}

```

```

13: import static ec.edu.espe.studentsystemcontroller.ClassroomController.readClassrooms;
14: import static ec.edu.espe.studentsystemcontroller.ClassroomController.validateActivityExt;
15: import static ec.edu.espe.studentsystemcontroller.PrintController;
16: import static ec.edu.espe.studentsystemcontroller.ThemesController;
17: import static ec.edu.espe.studentsystemcontroller.ThemesController.setFlatLightLaFTheme;
18: import static ec.edu.espe.studentsystemcontroller.Activity;
19: import static ec.edu.espe.studentsystemcontroller.Assignment;
20:
21: import java.util.EventQueue;
22: import java.util.concurrent.ConcurrentHashMap;
23: import java.util.Date;
24: import java.util.HashMap;
25: import java.util.List;
26: import java.util.Map;
27: import java.util.logging.Level;
28: import java.util.logging.Logger;
29: import java.util.Optional;
30: import java.util.SwingConstants;
31: import java.util.concurrent.TimeUnit;
32: import java.util.concurrent.TimeoutException;
33: import java.util.concurrent.atomic.AtomicInteger;
34: import java.util.concurrent.atomic.AtomicLong;
35: import java.util.concurrent.atomic.AtomicReference;
36:
37: package ec.edu.espe.studentsystem.view;
38:
39: import ec.edu.espe.studentsystemcontroller.ActivityController;
40: import ec.edu.espe.studentsystemcontroller.ActivityController.findActivities;
41: import ec.edu.espe.studentsystemcontroller.ActivityController.updateActivity;
42: import ec.edu.espe.studentsystemcontroller.ActivityController.updateGrade;
43: import ec.edu.espe.studentsystemcontroller.ClassroomController.findTeacher;
44: import ec.edu.espe.studentsystemcontroller.ClassroomController.getConnection;
45: import ec.edu.espe.studentsystemcontroller.PrintController;
46: import ec.edu.espe.studentsystemcontroller.ThemesController;
47: import static ec.edu.espe.studentsystemcontroller.ThemesController.setFlatLightLaFTheme;
48: import ec.edu.espe.studentsystemcontroller.Activity;
49: import ec.edu.espe.studentsystemcontroller.Assignment;
50: import java.util.EventQueue;
51: import java.util.concurrent.ConcurrentHashMap;
52: import java.util.Date;
53: import java.util.HashMap;
54: import java.util.List;
55: import java.util.Map;
56: import java.util.logging.Level;
57: import java.util.logging.Logger;
58: import java.util.Optional;
59: import java.util.SwingConstants;
60: import java.util.concurrent.TimeUnit;
61: import java.util.concurrent.TimeoutException;
62: import java.util.concurrent.atomic.AtomicInteger;
63: import java.util.concurrent.atomic.AtomicLong;
64: import java.util.concurrent.atomic.AtomicReference;

```

in form activity

```

25: package ec.edu.espe.studentsystem.view;
26:
27: import ec.edu.espe.studentsystemcontroller.ActivityController;
28: import ec.edu.espe.studentsystemcontroller.ActivityController.findActivities;
29: import ec.edu.espe.studentsystemcontroller.ActivityController.updateActivity;
30: import ec.edu.espe.studentsystemcontroller.ActivityController.updateGrade;
31: import ec.edu.espe.studentsystemcontroller.ClassroomController.findTeacher;
32: import ec.edu.espe.studentsystemcontroller.ClassroomController.getConnection;
33: import ec.edu.espe.studentsystemcontroller.PrintController;
34: import ec.edu.espe.studentsystemcontroller.ThemesController;
35: import static ec.edu.espe.studentsystemcontroller.ThemesController.setFlatLightLaFTheme;
36: import ec.edu.espe.studentsystemcontroller.Activity;
37: import ec.edu.espe.studentsystemcontroller.Assignment;
38: import java.util.EventQueue;
39: import java.util.concurrent.ConcurrentHashMap;
40: import java.util.Date;
41: import java.util.HashMap;
42: import java.util.List;
43: import java.util.Map;
44: import java.util.logging.Level;
45: import java.util.logging.Logger;
46: import java.util.Optional;
47: import java.util.SwingConstants;
48: import java.util.concurrent.TimeUnit;
49: import java.util.concurrent.TimeoutException;
50: import java.util.concurrent.atomic.AtomicInteger;
51: import java.util.concurrent.atomic.AtomicLong;
52: import java.util.concurrent.atomic.AtomicReference;

```

in classroom management.

lowerCamelCase in TeacherController

```

217: public static boolean deleteActivities(String classroomName, int teacherId) {
218:     MongoCollection activitiesCollection = getConnection().getCollection("activities");
219:     Bson filter = Filters.and(Filters.eq("teacherId", teacherId), Filters.eq("classroomName", classroomName));
220:     MongoCursor<Document> activities = activitiesCollection.find(filter).iterator();
221:     if (activities != null) {
222:         activitiesCollection.deleteMany(filter);
223:     }
224:     return true;
225: }
226:
227: public static boolean updateActivities(int teacherId, String nameToChange, String newName) {
228:     MongoCollection activitiesCollection = getConnection().getCollection("activities");
229:     Bson filter = Filters.and(Filters.eq("teacherId", teacherId), Filters.eq("classroomName", classroomName));
230:     MongoCursor<Document> activities = activitiesCollection.find(filter).iterator();
231:     if (activities != null) {
232:         Document updatedSubjectDocument = new Document("subjectName", newName);
233:         activitiesCollection.updateMany(filter, updatedSubjectDocument);
234:     }
235:     return true;
236: }

```

uncommunicative variable name in login controller

uncommunicative variable name in login controller

### set and get in Frm Student

in FrmLogin

## Team 10: ProgressTeam

Inspector: T9 Alex Trejo

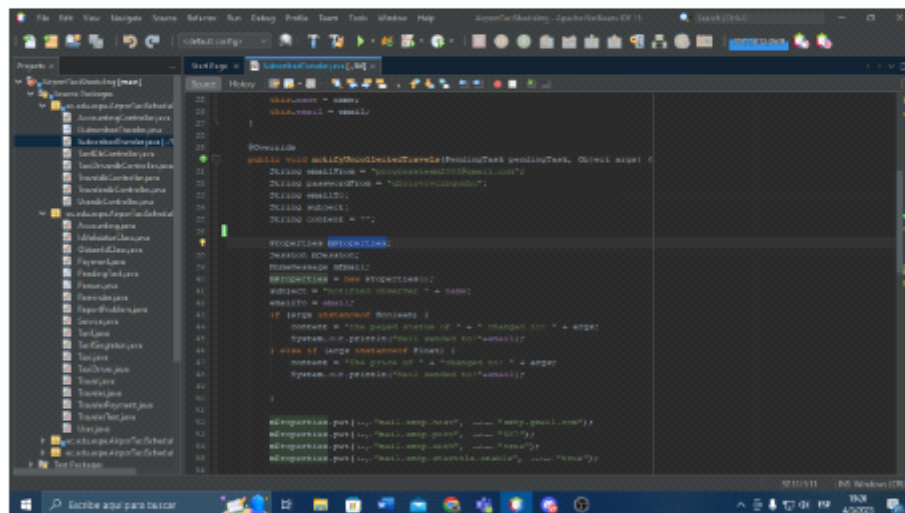
38 ULCO VELA SCO JEFFERSON MANUEL  
39 VERDUGO CABRERA WALTHER SEBASTIAN  
40 VILLARROEL BARRENO JUSTIN JOSHUA  
41 YARANGA SUQUILLO LEONARDO JAVIER

Code Quality -21/-20

- 7 Some names have a generic name ( mProperties, mSession, MimeMessage, mEmail, i,c,c)
- 7 Some variables are not used
- 1 Incorrect function name
- 5 Lack of identification in functions
- 1 A class that is not used in the code is imported

+3 For correcting view package variables

Total -21+21=0+3= +3



```
private void txtIdKeyTyped(java.awt.event.KeyEvent evt) {  
    char c = evt.getKeyChar();  
    if(c<'0' || c>'9'){  
        evt.consume();  
    }  
}
```

```
package ec.edu.espe.AirporTaxiScheduling.controller;  
  
import ec.edu.espe.AirporTaxiScheduling.controller.ISubscribedTraveler;  
import ec.edu.espe.AirporTaxiScheduling.model.PendingTask;  
import java.util.Properties;  
import javax.mail.Message;  
import javax.mail.MessagingException;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Session;  
import javax.mail.Transport;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;
```

```

9 import javax.mail.Session;
10 import javax.mail.Transport;
11 import javax.mail.internet.InternetAddress;
12 import javax.mail.internet.MimeMessage;
13 import javax.swing.JOptionPane;
14
15 /**
16  *
17  * @author Leonardo Yacanga, Progress Team, DCCO-ESPE
18  */
19 public class SubscribedTraveler implements ISubscribedTraveler {
20

```

