

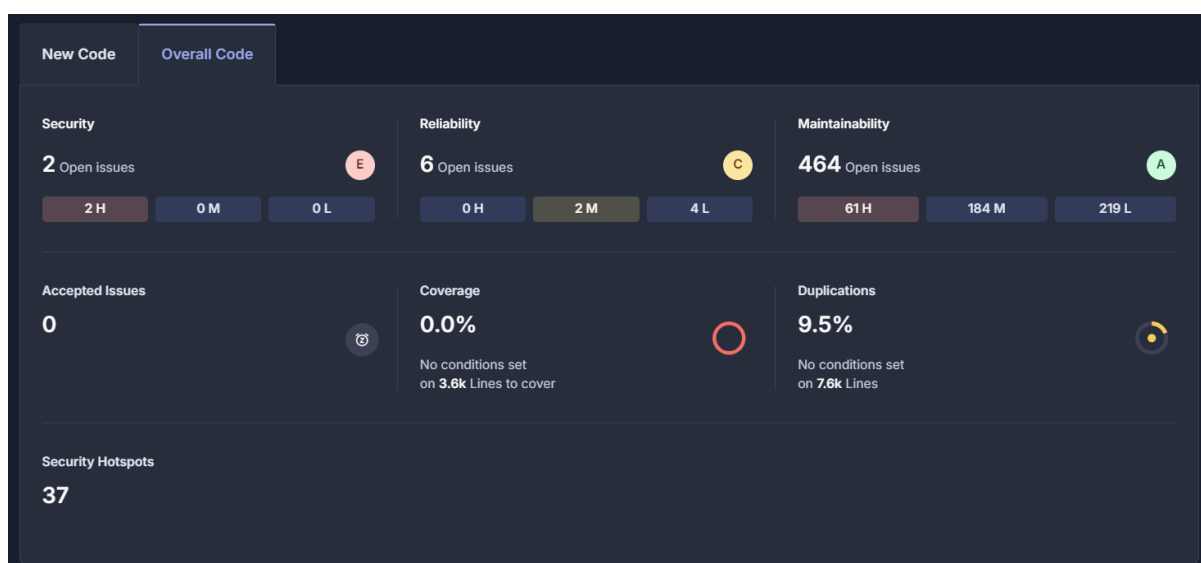
# SONNAR ISSUE ZUZENKETAK

9.taldea, Ibai Oñatibia, Iker Galarraga eta Jon Olea

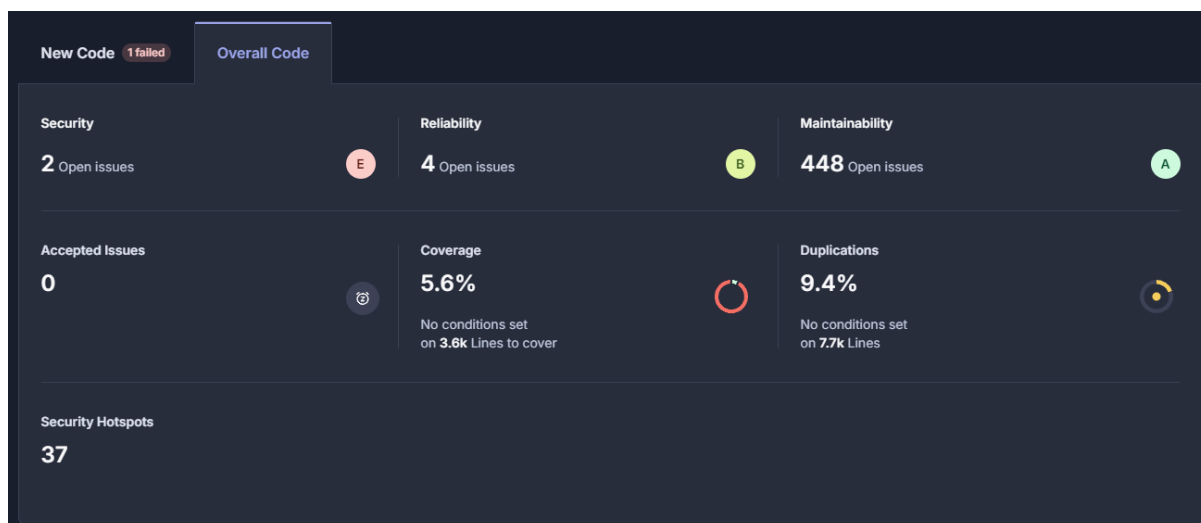
Proiektuaren sonnar link-a: <https://sonarcloud.io/project/overview?id=9-taldea-rides24>

Proiektuaren github link-a: <https://github.com/IkerGI28/Rides24E9>

Sonnar hasierako txostena



Sonnar amaierako txostena



Consistency

# major

## 1. Arazoa

src/.../domain/Driver.java

Replace the synchronized class "Vector" by an unsynchronized one such as "ArrayList" or "LinkedList".

```
@XmlIDREF
@OneToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
private List<Car> cars = new Vector<Car>();
@XmlIDREF
```

```
@XmlIDREF
@OneToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
private List<Car> cars = new ArrayList<>();
```

Arazoa konpontzeko, autoen bektorea erabili ordez, arrayList-a sortu dugu. Gainera, ArrayList-a "Car" motaz osatuta dagoela ez dugu jarri, "diamond operator" erabiliz.

## 2. Arazoa

src/main/java/domain/Traveler.java

Replace the synchronized class "Vector" by an unsynchronized one such as "ArrayList" or "LinkedList".

```
@XmlIDREF
@OneToMany(mappedBy = "traveler", fetch = FetchType.EAGER, cascade = CascadeType.PERSIST)
private List<Booking> bookedRides = new Vector<Booking>();
```

```
@XmlIDREF
@OneToMany(mappedBy = "traveler", fetch = FetchType.EAGER, cascade = CascadeType.PERSIST)
private List<Booking> bookedRides = new ArrayList<>();
```

Sinkronizazioa eraginkortasunari begira txarra denez, ohikoa da klase sinkronizatuen bertsio ez-sinkronizatuak erabiltzea. Horregatik hobe da erabiltzea LinkedList edo ArrayList klaseak Vector klasearen ordez.

## 3. Arazoa

src/main/java/domain/Traveler.java

Replace the synchronized class "Vector" by an unsynchronized one such as "ArrayList" or "LinkedList".

```
@XmlIDREF
@OneToMany(mappedBy = "traveler", fetch = FetchType.EAGER, cascade = CascadeType.PERSIST)
private List<Alert> alerts = new Vector<Alert>();
```

```
@XmlIDREF
@OneToMany(mappedBy = "traveler", fetch = FetchType.EAGER, cascade = CascadeType.PERSIST)
private List<Alert> alerts = new ArrayList<>();
```

Vector ordeaz ArrayList edo LinkedList erabiltzea gomendatzen du.

## minor

### 1. Arazoa

src/.../java/gui/BezeroGUI.java

Rename this local variable to match the regular expression `^[a-z][a-zA-Z0-9]*$`.

```
// lista
taula = new JTable();
List<Booking> Travellist = appFacadeInterface.getBookingFromDriver(username);
```

```
// lista
taula = new JTable();
List<Booking> travellist = appFacadeInterface.getBookingFromDriver(username);
```

Travellist izeneko lista “camelCase” formatura pasa dugu, travellist jarritz.

### 2. Arazoa

src/main/java/gui/AlertaAurkituakGUI.java

Rename this local variable to match the regular expression `^[a-z][a-zA-Z0-9]*$`

```
private void closeButton_actionPerformed(ActionEvent e) {
    this.setVisible(false);
}
```

```
private void closebutton() {
    this.setVisible(false);
}
```

“\_” duten izenak letra larriz izatea gomendatzen duenez izena moztu egin dut, horrez gain erabiltzen ez den e atributua ezabatu dut.

### 3. Arazoa

src/main/java/gui/MovementsGUI.java

Rename this method name to match the regular expression `^[a-z][a-zA-Z0-9]*$`.

```
private void iButtonClose_actionPerformed(ActionEvent e) {
    this.setVisible(false);
}
```

```
private void botoiaitxi() {
    this.setVisible(false);
}
```

Gomendatzen da metodoaren izena letra larritz idaztea '\_' karakterearekin, letra xehez idatzita eta '\_' karaktererik gabe idatzi dut eta e parametroa kendu diot metodoari.

## Intentionality

### major

#### 1. Arazoa

src/main/java/domain/Driver.java

Strings and Boxed types should be compared using "equals()".

```
if ((ride.getFrom()==from) && (ride.getTo()==to) && (ride.getDate().equals(date))) {
    //ride = createRide();
    if ((ride.getFrom().equals(from)) && (ride.getTo().equals(to)) && (ride.getDate().equals(date))) {
```

'=='-k ez ditu bi aldagaien balioak konparatzen baizik eta bien memoria posizioak konparatzen ditu, ondorioz gomendagarriagoa da .equals() metodoa erabiltzea.

#### 2. Arazoa

src/main/java/businessLogic/BLFacade.java

This block of commented-out lines of code should be removed

```
6 import java.util.List;
7 |
8 //import domain.Booking;
9 import domain.Ride;
10 import domain.Traveler;
11 import domain.User;
12 //import domain.Admin;
13 import domain.Alert;

3•import java.util.Date;
4 import java.util.List;
5 import domain.Ride;
6 import domain.Traveler;
7 import domain.User;
8 import domain.Alert;
```

Kodigo garbia mantentzeko komentario horiek ezabatzea lagungarria da.

#### 3. Arazoa

src/.../domain/Driver.java

Add the "@Override" annotation above this method signature

```
public String toString() {
    return (super.toString());
}
```

```
@Override
public String toString() {
    return (super.toString());
}
```

@Override gehitu diogu Driver-en toString metodoari.

## minor

### 1. Arazoa

src/main/java/configuration/ConfigXML.java

Remove this unnecessary cast to "Node".

```
private static String getTagValue(String sTag, Element eElement)
{
    NodeList nlList= eElement.getElementsByTagName(sTag).item(0).getChildNodes();
    Node nValue = (Node) nlList.item(0);

    return nValue.getNodeValue();
}
```

```
private static String getTagValue(String sTag, Element eElement)
{
    NodeList nlList= eElement.getElementsByTagName(sTag).item(0).getChildNodes();
    Node nValue = nlList.item(0);

    return nValue.getNodeValue();
}
```

Ez da casting-a egin behar Node klaseko aldagaia erazagutzean esplizituki esaten ari garelako zein klasekoa izan behar den objektua.

### 2. Arazoa

src/main/java/businessLogic/BLFacadeImplementation.java

Remove this empty statement.

```
@WebMethod
public Ride createRide(String from, String to, Date date, int nPlaces, float price, String driverName)
    throws RideMustBeLaterThanTodayException, RideAlreadyExistException {

    dbManager.open();
    Ride ride = dbManager.createRide(from, to, date, nPlaces, price, driverName);
    dbManager.close();
    return ride;
};
```

```
@WebMethod
public Ride createRide(String from, String to, Date date, int nPlaces, float price, String driverName)
    throws RideMustBeLaterThanTodayException, RideAlreadyExistException {}

    dbManager.open();
    Ride ride = dbManager.createRide(from, to, date, nPlaces, price, driverName);
    dbManager.close();
    return ride;
}
```

Beharrezkoak ez diren ; ezabatu behar dira al den garbiena gelditzeko.

### 3. Arazoa

src/.../DataAccess.java

Complete the task associated to this TODO comment.

```
        return ride;
    } catch (NullPointerException e) {
        // TODO Auto-generated catch block
        return null;
    }
```

```
        return ride;
    } catch (NullPointerException e) {
        e.printStackTrace();
        return null;
    }
```

Arazoa, TODO komentarioa oraindik kendu gabe zegoelako da, beraz, komentarioa kendu, eta errore kasuan errorea kontsolan pantailaratzeko ere gehitu diogu.

## critical

### 1. Arazoa

src/main/java/gui/BaloraGUI.java

Use static access with "javax.swing.SwingConstants" for "HORIZONTAL".

```
slider = new JSlider(JSlider.HORIZONTAL, 1, 5, 1);
```

```
slider = new JSlider(SwingConstants.HORIZONTAL, 1, 5, 1);
```

Aldagai estatikoak klasearekin daude lotuta ez instantzia zehatz baten bidez, horregatik metodo bati deitzerakoan hobe da oinarrizko klase estatikoaren bidez deia egitea eta ez instantziaren bidez.

## 2. Arazoa

src/main/java/businessLogic/BusinessLogicServer.java

Use static access with "javax.swing.WindowConstants" for "DISPOSE\_ON\_CLOSE".

```
public static void main(String[] args) {  
    try {  
        BusinessLogicServer dialog = new BusinessLogicServer();  
        dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);  
        dialog.setVisible(true);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

```
public static void main(String[] args) {  
    try {  
        BusinessLogicServer dialog = new BusinessLogicServer();  
        dialog.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);  
        dialog.setVisible(true);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

Erroreak adierazten duen bezala JDialog ordeztu WindowConstants erabiltze gomendatzen da.

## 3. Arazoa

src/.../java/gui/KotxeaGehituGUI.java

Use static access with "javax.swing.WindowConstants" for "EXIT\_ON\_CLOSE".(jon)

```
public KotxeaGehituGUI(String username) {  
  
    KotxeaGehituGUI.setBussinessLogic(MainGUI.getBusi  
  
    this.getContentPane().setLayout(null);  
    this.setSize(new Dimension(400, 250));  
    this.setTitle(ResourceBundle.getBundle("Etiquetas  
    this.setResizable(false);  
  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```

public KotxeaGehituGUI(String username) {

    KotxeaGehituGUI.setBussinessLogic(MainGUI.getBusinessLogi

    this.getContentPane().setLayout(null);
    this.setSize(new Dimension(400, 250));
    this.setTitle(ResourceBundle.getBundle("Etiquetas").getSt
    this.setResizable(false);

    setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);

```

JFrame barruko EXIT\_ON\_CLOSE konstantea erabili orde, WindowConstants barruan dagoen EXIT\_ON\_CLOSE konstantea erabili dugu, arazoak aipatzen zuen bezala.

## Adaptability

### major

Replace this use of System.out by a logger.

#### 1. Arazoa

src/main/java/configuration/ConfigXML.java

Replace this use of System.out by a logger.

```

} catch (Exception e) {
    System.out.println("Error in ConfigXML.java: problems with " + configFile);
    e.printStackTrace();
}

```

```

transient Logger logger = Logger.getLogger(getClass().getName());

```

```

} catch (Exception e) {
    logger.info(String.Format("Error in ConfigXML.java: problems with " + configFile));
    e.printStackTrace();
}

```

#### 2. Arazoa

src/main/java/dataAccess/ObjectdbManagerServer.java

Replace this use of System.out by a logger.



```
okButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textArea.append("\n\n\nClosing the database... ");
        try {
            System.out.println("Server close");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
});
```

```
okButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textArea.append("\n\n\nClosing the database... ");
        try {
            Logger logger = Logger.getLogger(getClass().getName());
            logger.info("Server close");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
});
```

### 3. Arazoa

src/.../domain/Driver.java

Replace this use of System.out by a logger.

```
if (encontrado) {
    System.out.println("posicion "+ pos);
    r=createdRides.get(pos);
}
```

```
transient Logger logger = Logger.getLogger(getClass().getName());
```

```
if (logger.isLoggable(Level.INFO)) {
    logger.info(String.format("Position: %d", pos));
}
```

Arazoa konpontzeko, System.out.println() erabili ordez kontsolan gauzak idazteko, logger bat erabili behar izan dugu. Lehenengo logger bat deklaratu dugu, transient erabiliz (logger-a ez serializatzeke).

Ondoren, logger-aren info erabili ahal bada, mezua kontsolan idatziko du, mezu formatea erabiliz, posizioa adierazteko.

## minor

### 1. Arazoa

src/test/java/CreateRideBDBlackTest.java

Move this file to a named package.

```

> (default package)
> CreateRideBDBlackTest.java
```

```
mocktest
CreateRideBDBlackTest.java
```

Izena ez daukan pakete bat egotea proiektu batean ideia ona ez denez, gomendagarria da klase eta fitxategi guztiak koherentziazko pakete izendunetan jasotzea.

## 2. Arazoa

src/.../java/gui/KotxeaGehituGUI.java

```
} catch (RideMustBeLaterThanTodayException e1) {
    jLabelMsg.setText(e1.getMessage());
} catch (RideAlreadyExistException e1) {
    jLabelMsg.setText(e1.getMessage());
}
```

```
} catch (RideMustBeLaterThanTodayException | RideAlreadyExistException e1) {
    jLabelMsg.setText(e1.getMessage());
}
```

Bi catch-ek berdina egiten zutenez, testu berdina aldatu, biak catch bakarrean elkartu ditugu.

## 3. Arazoa

src/main/java/domain/Complaint.java

Make egoera a static final constant or non-public and provide accessors if needed.

```
public Boolean aurkeztua;
public String egoera;
```

```
private Boolean aurkeztua;
private String egoera;
```

Egoera atributo publikoa zen eta estatikoan edo ez-publikoan ezartzea eskatzen du, ondorioz atributua pribatuan ezarri dugu, ez delako beharrezkoa publikoa izatea. aurkezpena atributuak ere arazo berdina zuenenez biak aldatu ditugu.

## critical

### 1.Arazoa

src/main/java/dataAccess/DataAccess.java

Define a constant instead of duplicating this literal "Madrid" 3 times

```
driver1.addRide("Donostia", "Madrid", date2, 5, 20); //ride1
driver1.addRide("Irun", "Donostia", date2, 5, 2); //ride2
driver1.addRide("Madrid", "Donostia", date3, 5, 5); //ride3
driver1.addRide("Barcelona", "Madrid", date4, 0, 10); //ride4
driver2.addRide("Donostia", "Hondarribi", date1, 5, 3); //ride5
```

Madrid ez errepikatzeko 3 aldiz "Madrid" string-a duen konstante bat erazagutu dugu.

```
final String MADRID = "Madrid";
driver1.addRide("Donostia", MADRID, date2, 5, 20); //ride1
driver1.addRide("Irun", "Donostia", date2, 5, 2); //ride2
driver1.addRide(MADRID, "Donostia", date3, 5, 5); //ride3
driver1.addRide("Barcelona", MADRID, date4, 0, 10); //ride4
driver2.addRide("Donostia", "Hondarribi", date1, 5, 3); //ride5
```

Define a constant instead of duplicating this literal "NotCompleted" 3 times.(Ibai)

## 2.Arazoa

src/main/java/dataAccess/DataAccess.java

Define a constant instead of duplicating this literal "Donostia" 4 times

```
driver1.addRide("Donostia", "Madrid", date2, 5, 20); //ride1
driver1.addRide("Irun", "Donostia", date2, 5, 2); //ride2
driver1.addRide("Madrid", "Donostia", date3, 5, 5); //ride3
driver1.addRide("Barcelona", "Madrid", date4, 0, 10); //ride4
driver2.addRide("Donostia", "Hondarribi", date1, 5, 3); //ride5
```

```
final String DONOSTIA = "Donostia";
```

```
driver1.addRide(DONOSTIA, "Madrid", date2, 5, 20); //ride1
driver1.addRide("Irun", DONOSTIA, date2, 5, 2); //ride2
driver1.addRide("Madrid", DONOSTIA, date3, 5, 5); //ride3
driver1.addRide("Barcelona", "Madrid", date4, 0, 10); //ride4
driver2.addRide(DONOSTIA, "Hondarribi", date1, 5, 3); //ride5
```

4 Aldiz String bera jarri ordeaz, "Donostia" duen konstante bat sortu dugu eta jarri dugu.

## 3.Arazoa

src/main/java/dataAccess/DataAccess.java

Define a constant instead of duplicating this literal "BookFreeze" 5 times.

```
Movement m1 = new Movement(traveler1, "BookFreeze", 20);
Movement m2 = new Movement(traveler1, "BookFreeze", 40);
Movement m3 = new Movement(traveler1, "BookFreeze", 5);
Movement m4 = new Movement(traveler2, "BookFreeze", 4);
Movement m5 = new Movement(traveler1, "BookFreeze", 3);
Movement m6 = new Movement(driver1, "Deposit", 15);
Movement m7 = new Movement(traveler1, "Deposit", 160);
```

```
final String BOOKFREEZE = "BookFreeze";

Movement m1 = new Movement(traveler1, BOOKFREEZE, 20);
Movement m2 = new Movement(traveler1, BOOKFREEZE, 40);
Movement m3 = new Movement(traveler1, BOOKFREEZE, 5);
Movement m4 = new Movement(traveler2, BOOKFREEZE, 4);
Movement m5 = new Movement(traveler1, BOOKFREEZE, 3);
Movement m6 = new Movement(driver1, "Deposit", 15);
Movement m7 = new Movement(traveler1, "Deposit", 168);
```

bookFreeze Atributua bost aldiz ezarri ordez atributu batean txertatzea eta hau erabiltzea gomendatzen da.

## Responsibility

**major**

-

**minor**

-

**critical**

Make sure this SonarQube token gets revoked, changed, and removed from the code.

```
<properties>
  <maven.compiler.target>1.8</maven.compiler.target>
  <maven.compiler.source>1.8</maven.compiler.source>
  <sonar.organization>pokoion</sonar.organization>
  <sonar.host.url>https://sonarcloud.io</sonar.host.url>
  <sonar.projectKey>9-taldea-rides24</sonar.projectKey>
  <sonar.login>9960376f047f7fde48fec0b8afaad70402eeb532</sonar.login>
</properties>
```

Arazo hau ez dugu zuzendu, ez genekien non jarri token-a