

Class Printer: Has single responsibility. only create a format.

```
public Printer(JTable table) {
    this.table = table;
}

@Override
public int print(Graphics graphics, PageFormat pageFormat, int pageIndex) throws PrinterException {
    if (pageIndex == 0) {
        Graphics2D graphics2d = (Graphics2D) graphics;
        graphics2d.translate((x: pageFormat.getImageableX(), y: pageFormat.getImageableY()));

        // Imprimir el contenido de la tabla
        table.print(g, graphics2d);

        return Printable.PAGE_EXISTS;
    } else {
        return Printable.NO_SUCH_PAGE;
    }
}
}
```

Class MongoDBConnection: Maybe has single responsibility.

```
public static Object[][] generateTableData(String collectionName, String[] fields) {
    List<Object[]> dataList = new ArrayList<>();
    try {
        MongoClient mongoClient = MongoClient.create("mongodb+srv://jseanmartin:12345@managementsystem.kk");
        MongoDB database = mongoClient.getDatabase("SchoolManagementSystem");
        MongoCollection<Document> collection = database.getCollection(collectionName);
        FindIterable<Document> result = collection.find();

        for (Document document : result) {
            Object[] rowData = new Object[fields.length];
            for (int i = 0; i < fields.length; i++) {
                rowData[i] = document.get(fields[i]);
            }
            dataList.add(rowData);
        }
    }
    Object[][] data = new Object[dataList.size()][fields.length];

    for (int i = 0; i < dataList.size(); i++) {
        data[i] = dataList.get(i);
    }
    return data;
}
```

FrmUpdatePaymentStudent: Has single responsibility.

```
private void btnDeletePaidActionPerformed(java.awt.event.ActionEvent evt) {

    String id = txtIdStudent.getText().trim();
    String idValid = validId(cedula: id);
    showAlert(idValid, id);

}

private void txtIdStudentActionPerformed(java.awt.event.ActionEvent evt) {
    Validation validation = new Validation();
    validation.validateNumber(textField: txtIdStudent);
}
```

FrmPrintDataPayments: Has single responsibility

```
private void btnPrintActionPerformed(java.awt.event.ActionEvent evt) {
    printTable(table: jTable1);
}
}
```

PaymentRecord Has single responsibility and open-close

```
public void deletePaid(String studentId) {
    MongoCollection<Document> PaymentCollection = mdb.getCollection();
    Document student = PaymentCollection.find(new Document("_id", studentId)).first();
    if (student != null) {
        Number monthlyPaymentNumber = (Number) student.get("monthlyPayment");
        Double monthlyPayment = monthlyPaymentNumber.doubleValue();

        Document filter = new Document("_id", studentId);
        Document update = new Document("$set", new Document("valuePaid", 0.0).append("remainingValue", monthlyPayment));

        PaymentCollection.updateOne(filter, update);
    }
}
```

The view models has single responsibility the buttons only do what they do.  
in the controller models maybe some clases has single responsibility.