Exp Name: Cleaning Excel File Using Apache POI

Namansh Singh Maurya
22MIA1034

Aim

To develop a Java application using the Apache POI library that cleans an Excel file by removing invalid data and duplicate entries, improving data quality and integrity.

Algorithm

1. File Initialization
    - Define input and output file paths
    - Create FileInputStream for reading the source Excel file
    - Initialize Apache POI Workbook objects for input and output
2. Data Processing
    - Iterate through each row and cell in the source worksheet
    - Apply cleaning operations:
        - Remove leading and trailing whitespace from text
        - Validate cell contents
        - Handle different data types (string, numeric, boolean)
    - Skip rows containing empty or invalid data
    - Track and remove duplicate entries
3. Output Generation
    - Write cleaned data to a new workbook
    - Save the processed data to the output file
    - Clean up resources and close file streams

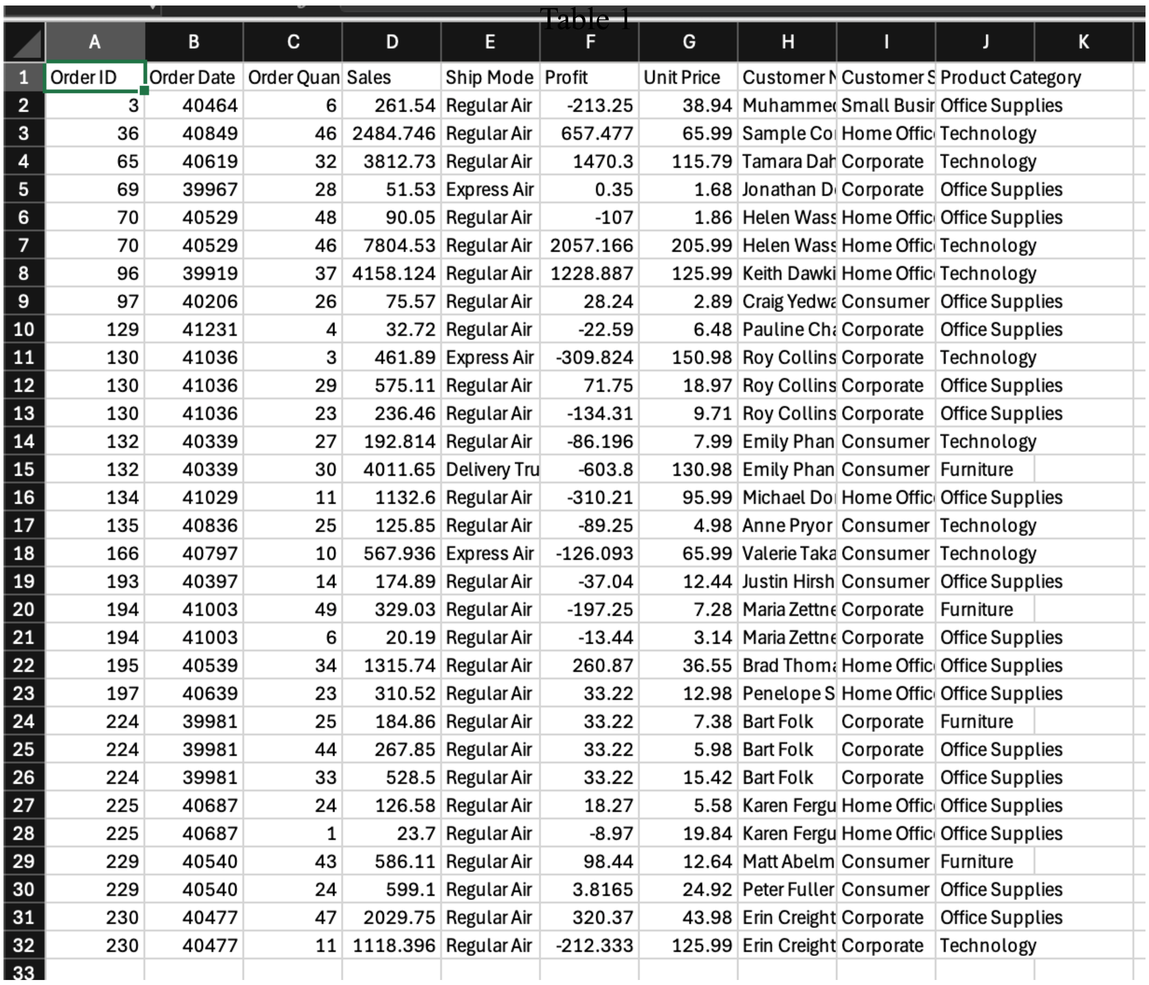
Program Implementation

```
package com.hp;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellType;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
```

```
public class ExcelDataCleaner {
   public static void main(String[] args) {
      String sourceFile = "/Users/HP/Documents/bigdataframeworks/Table 2.xlsx";
      String destinationFile = "/Users/HP/Documents/bigdataframeworks/Cleaned_Table
2.xlsx";

      try (FileInputStream fileIn = new FileInputStream(sourceFile);
          Workbook originalWorkbook = new XSSFWorkbook(fileIn);
          Workbook processedWorkbook = new XSSFWorkbook()) {

         Sheet originalSheet = originalWorkbook.getSheetAt(0);
         Sheet processedSheet =
processedWorkbook.createSheet(originalSheet.getSheetName());

         int processedRowCount = 0;

         for (Row currentRow : originalSheet) {
            Row newRow = processedSheet.createRow(processedRowCount);
            boolean isRowValid = true;
            List<Object> processedData = new ArrayList<>();

            for (Cell currentCell : currentRow) {
               Cell newCell = newRow.createCell(currentCell.getColumnIndex());
               switch (currentCell.getCellType()) {
                  case STRING:
                     String cellValue = currentCell.getStringCellValue().trim();
                     if (cellValue.isEmpty()) {
                        isRowValid = false;
                     }
                     newCell.setCellValue(cellValue);
                     processedData.add(cellValue);
                     break;

                  case NUMERIC:
                     newCell.setCellValue(currentCell.getNumericCellValue());
                     processedData.add(currentCell.getNumericCellValue());
                     break;

                  case BOOLEAN:
                     newCell.setCellValue(currentCell.getBooleanCellValue());
                     processedData.add(currentCell.getBooleanCellValue());
                     break;

                  default:
                     newCell.setCellValue("N/A");
                     isRowValid = false;
                     processedData.add("N/A");
               }
            }
            if (processedData.contains(null) || processedData.contains("")) {
                isRowValid = false;
            }

            if (isRowValid) {
                processedRowCount++;
            } else {
                processedSheet.removeRow(newRow);
            }
        }

        removeDuplicateRows(processedSheet);

        try (FileOutputStream fileOut = new FileOutputStream(destinationFile)) {
            processedWorkbook.write(fileOut);
        }

        System.out.println("Data cleaning completed. Cleaned data saved to " +
destinationFile);

    } catch (Exception e) {
        e.printStackTrace();
    }
  }

  private static void removeDuplicateRows(Sheet sheet) {
    Set<String> uniqueRowContents = new HashSet<>();
    List<Integer> rowsForDeletion = new ArrayList<>();

    for (Row currentRow : sheet) {
        StringBuilder rowContent = new StringBuilder();

        for (Cell cell : currentRow) {
            if (cell.getCellType() == CellType.STRING) {
                rowContent.append(cell.getStringCellValue()).append("|");
            } else if (cell.getCellType() == CellType.NUMERIC) {
                rowContent.append(cell.getNumericCellValue()).append("|");
            }
        }

        if (!uniqueRowContents.add(rowContent.toString())) {
            rowsForDeletion.add(currentRow.getRowNum());
        }
    }

    for (int i = rowsForDeletion.size() - 1; i >= 0; i--) {
        int rowIndex = rowsForDeletion.get(i);
        sheet.removeRow(sheet.getRow(rowIndex));
        sheet.shiftRows(rowIndex + 1, sheet.getLastRowNum(), -1);
        }
    }
}
```

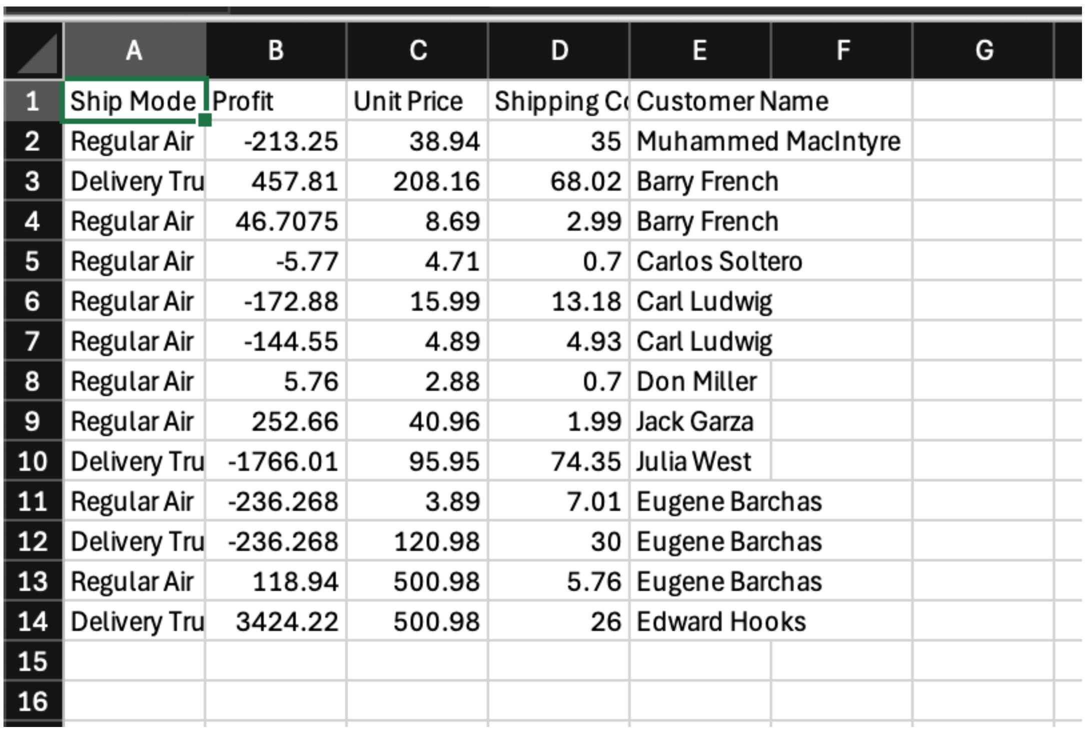
Output

Table 1



	A	B	C	D	E	F	G	H	I	J	K
1	Order ID	Order Date	Order Quan	Sales	Ship Mode	Profit	Unit Price	Customer N	Customer S	Product Category	
2	3	40464	6	261.54	Regular Air	-213.25	38.94	Muhammed	Small Busir	Office Supplies	
3	36	40849	46	2484.746	Regular Air	657.477	65.99	Sample Co	Home Offic	Technology	
4	65	40619	32	3812.73	Regular Air	1470.3	115.79	Tamara Dah	Corporate	Technology	
5	69	39967	28	51.53	Express Air	0.35	1.68	Jonathan D	Corporate	Office Supplies	
6	70	40529	48	90.05	Regular Air	-107	1.86	Helen Wass	Home Offic	Office Supplies	
7	70	40529	46	7804.53	Regular Air	2057.166	205.99	Helen Wass	Home Offic	Technology	
8	96	39919	37	4158.124	Regular Air	1228.887	125.99	Keith Dawki	Home Offic	Technology	
9	97	40206	26	75.57	Regular Air	28.24	2.89	Craig Yedwa	Consumer	Office Supplies	
10	129	41231	4	32.72	Regular Air	-22.59	6.48	Pauline Cha	Corporate	Office Supplies	
11	130	41036	3	461.89	Express Air	-309.824	150.98	Roy Collins	Corporate	Technology	
12	130	41036	29	575.11	Regular Air	71.75	18.97	Roy Collins	Corporate	Office Supplies	
13	130	41036	23	236.46	Regular Air	-134.31	9.71	Roy Collins	Corporate	Office Supplies	
14	132	40339	27	192.814	Regular Air	-86.196	7.99	Emily Phan	Consumer	Technology	
15	132	40339	30	4011.65	Delivery Tru	-603.8	130.98	Emily Phan	Consumer	Furniture	
16	134	41029	11	1132.6	Regular Air	-310.21	95.99	Michael Do	Home Offic	Office Supplies	
17	135	40836	25	125.85	Regular Air	-89.25	4.98	Anne Pryor	Consumer	Technology	
18	166	40797	10	567.936	Express Air	-126.093	65.99	Valerie Taka	Consumer	Technology	
19	193	40397	14	174.89	Regular Air	-37.04	12.44	Justin Hirsh	Consumer	Office Supplies	
20	194	41003	49	329.03	Regular Air	-197.25	7.28	Maria Zettne	Corporate	Furniture	
21	194	41003	6	20.19	Regular Air	-13.44	3.14	Maria Zettne	Corporate	Office Supplies	
22	195	40539	34	1315.74	Regular Air	260.87	36.55	Brad Thoma	Home Offic	Office Supplies	
23	197	40639	23	310.52	Regular Air	33.22	12.98	Penelope S	Home Offic	Office Supplies	
24	224	39981	25	184.86	Regular Air	33.22	7.38	Bart Folk	Corporate	Furniture	
25	224	39981	44	267.85	Regular Air	33.22	5.98	Bart Folk	Corporate	Office Supplies	
26	224	39981	33	528.5	Regular Air	33.22	15.42	Bart Folk	Corporate	Office Supplies	
27	225	40687	24	126.58	Regular Air	18.27	5.58	Karen Fergu	Home Offic	Office Supplies	
28	225	40687	1	23.7	Regular Air	-8.97	19.84	Karen Fergu	Home Offic	Office Supplies	
29	229	40540	43	586.11	Regular Air	98.44	12.64	Matt Abelm	Consumer	Furniture	
30	229	40540	24	599.1	Regular Air	3.8165	24.92	Peter Fuller	Consumer	Office Supplies	
31	230	40477	47	2029.75	Regular Air	320.37	43.98	Erin Creight	Corporate	Office Supplies	
32	230	40477	11	1118.396	Regular Air	-212.333	125.99	Erin Creight	Corporate	Technology	
33											

Table 2



	A	B	C	D	E	F	G
1	Ship Mode	Profit	Unit Price	Shipping Co	Customer Name		
2	Regular Air	-213.25	38.94	35	Muhammed MacIntyre		
3	Delivery Tru	457.81	208.16	68.02	Barry French		
4	Regular Air	46.7075	8.69	2.99	Barry French		
5	Regular Air	-5.77	4.71	0.7	Carlos Soltero		
6	Regular Air	-172.88	15.99	13.18	Carl Ludwig		
7	Regular Air	-144.55	4.89	4.93	Carl Ludwig		
8	Regular Air	5.76	2.88	0.7	Don Miller		
9	Regular Air	252.66	40.96	1.99	Jack Garza		
10	Delivery Tru	-1766.01	95.95	74.35	Julia West		
11	Regular Air	-236.268	3.89	7.01	Eugene Barchas		
12	Delivery Tru	-236.268	120.98	30	Eugene Barchas		
13	Regular Air	118.94	500.98	5.76	Eugene Barchas		
14	Delivery Tru	3424.22	500.98	26	Edward Hooks		
15							
16							

Result

The experiment successfully demonstrates the ability to clean an Excel file by removing invalid entries and duplicates using Apache POI in Java.