# Dataset Exercise

**Exercise**

Company X recently contacted Cost Engineering as they are interested in managing their data in Cleopatra Enterprise. They would like to estimate the cost of their construction project through the cumulative calculation of smaller parts such as pipes, cables etc. To achieve this, they would like Cost Engineering to store and process their project data. The current data is stored in a simple **CSV file** that uses a **semicolon** as separator, a **period** to indicate decimal digits and **no symbol** to separate thousands (e.g. 42956.15).  Each line indicates either a cost object's details and its cost, or external costs that need to be taken into account (both would be referred to as "cost item"). Each line will have the fields **Name**, **Type**, **Value** and **GroupingKey**. The type of an item refers to the previous distinction, and grouping key is important to distinguish the cost of different categories. You can assume that the fields are non-empty/non-null and case-sensitive.

We would like you to create a Java application for this objective. The application should:

* Allow to read an input file's items in the format explained in detail below and store the read data in an appropriate manner
* Allow the user to pose queries through the application's user interface to find and work with the data. You can use any **Java-based UI** framework you want to use for the application (Swing, JavaFX etc.), but the use of UI Generators/GUI Builders are not allowed (e.g. *SceneBuilder*). We want the user to be able to use the following queries:
  + Total direct cost
  + Total indirect cost
  + Total cost (total direct + total indirect cost)
  + Total direct cost for grouping key X
  + Total indirect cost for grouping key X
  + Total cost for grouping key X

The ideal solution should be maintainable (easy to modify) and scalable/extendable (what if another cost type were to be added?). You are free to design your own application structure as long as the previously mentioned functionalities are present. Please estimate the time it would take for you to finish the exercise, and also track the time you have spent on the exercise. We expect that you spend between 4 and 8 hours on the exercise.

For this exercise you have been given an example UI layout for reference and calculations with an example input file. You can check the formatting and the example input/result below for further details.

**Data Format**

|  |  |
| --- | --- |
| **Column name** | **Explanation** |
| NAME | Name of the cost item. |
| TYPE | Type of the cost item. It can be DIRECT or INDIRECT, indicating whether it's an object or an external cost. |
| VALUE | item's type is DIRECT -> a decimal value in the range [0.0, MAX\_DOUBLE]. This value represents the real cost of the object.  item's type is INDIRECT > a decimal value in the range [0.0, 100.0]. This value represents a percentage of the total cost assigned to a GROUPING\_KEY. |
| GROUPING\_KEY | Always filled in. Represents the categorization of cost items. An item belongs to exactly one grouping key. |

**Example UI**

Graphical user interface

Description automatically generated with medium confidence

**Example Calculation *(see resources/ExerciseCSV.csv)***  
Cable installation;DIRECT;500;Installation  
Pipe installation;DIRECT;700;Installation  
Maintenance;INDIRECT;10.0;Installation   
120 sq-mm Copper Power Cable;DIRECT;400;Electrical  
Pipe SS 12 inch;DIRECT;200;Electrical  
Material loss;INDIRECT;5.0;Electrical  
Class 7 Inch PVC Pipe;DIRECT;100;Heating  
Fiberglass Pipe Insulation;DIRECT; 200;Heating  
  
**Expected Result**  
Total direct cost: 500+700+400+200+100+200=2100  
Total indirect cost: (400+200)\*0.05 (electrical)+(500+700)\*0.10 (installation)=150  
Total cost: 2100+150=2250  
Total direct cost for grouping key "Electrical":  400+200=600  
Total indirect cost for grouping key "Electrical": 600\*0.05=30  
Total cost for grouping key  "Electrical": 600+30=630