Formal Languages and Compilers – JFlex & Cup

Using the JFLEX lexer generator and the CUP parser generator, realize a Java program that recognizes a language describing food prices and shop lists and capable of computing the prices of purchased food.

Input language

The input file is divided in 2 sections by means of "%%" (2 percent symbols).

The first section comprises a non empty list of food prices. Each element of the list has the following fields and is terminated by a semicolon:

<food ID> <price in Euro> "euro/hg" ;

<food ID> is an alphanumeric string starting with a letter and containing letters, digits and underscores.
< price in Euro > is a decimal number with 2 digit precision for the fractional.
The food ID should be used as a key in a Symbol Table in order to retrieve the relative price.

The second section is started by a date and time information that is is prefixed by the keyword "date" followed by a colon character, ":" and is terminated by a semicolon symbol, ";". Between the prefix and the terminal symbol date and time information are expressed:

- The date is expressed in the format YYYY.MM.DD where DD is a number between 01 and 31 and MM is a number between 01 and 12; YYYY is a number between 1000 and 9999.
- The time element is OPTIONAL and is enclosed by the symbols "(" and ")"; it is expressed with the format HH:MM where HH is a number between 00 and 11 and MM is a number between 00 and 59.

(Number formats of date and time should be managed in the scanner by means of regular expressions) Date and time elements can be in any order; all the following writings are then valid:

date : 2007.06.11 (18:30) ;
date: (18:30) 2008.01.01 ;
date: 2008.09.09 ;

After the date token there is a non empty list of purchases associated to groups of buyers.
Each purchase is on the same line and is terminated by means of a semicolon, ";". Each purchase follows the following syntax:
<number of people> "people" ":" <group ID> ":" <shop list>
Where:

- <number of people> is an integer value.
- <group ID> is an alphanumeric string starting with an uppercase letter and containing letters, digits or underscores "_".
- <shop list> is a list of at least 2 elements separated by means of a comma, "," and where each element follows this syntax:
  - <quantity> "hg" <food ID>, where <quantity> is an integer and <food ID> is one of the ID specified in the first section of the input source file. A given <food ID> can appear at most once per purchase.

Each groups share the expenses for its own shop list. Each group can appear in the list at most once.

Objective

The program should validate the input file against the language previously described and if the input is in a correct syntax the program should compute the amount of money expended by each group and totally. The report on the expenses should detail the expenses for different component (the total and the fare of each member) and alse report the overall expenses for the group purchase (the total and the total fare for each group member). See details on the example below.
Te program should also compute the total of all the shop lists present in the input file.

The only global variable allowed in the parser program is the symble table for mapping food IDs to prices. In order to compute the member fares of group expenses the information on the number of people in the group MUST be accessed by means of inherited attributes.
Solutions using global variables are not accepted. Syntetized attributes and predefined object RESULT should be used for propagating the information.

Example:

INPUT

```
cheese 3.00 euro/hg ;
HAM 2.00 euro/hg ;
HAM_deluxe 4.00 euro/hg ;
Bread 1.50 euro/hg ;

%%
date: (12:00) 2008.06.06 ;
2 people : Group1 : 4 hg HAM_deluxe, 6 hg Bread , 4 hg cheese;
2 people : Group_2 : 8 hg Bread, 6 hg cheese;
4 people : GR3 : 10 hg HAM, 7 hg cheese, 11 hg Bread;
```

OUTPUT

```
HAM_deluxe : total 16 euro - splitted 8 euro
Bread: total 9 euro – splitted 4.5 euro
Cheese: total 12 euro – splitted 6 euro
Overall: total 37 euro – splitted 18,5 euro
Group1 (2 people):

---
Bread: total 12 euro – splitted 6 euro
Cheese: total 18 euro – splitted 9 euro
Overall: total 30 euro – splitted 15 euro
Group_2 (2 people):

---
HAM: total 20 euro – splitted 5 euro
Bread: total 16,5 euro – splitted 4,125 euro
Cheese: total 21 euro – splitted 5,25 euro
Overall: total 57,5 euro – splitted 14,375 euro
GR3 (4 people)

---
TOTAL 114 euro
```