Final Project

Software Development

Universidad Carlos III de Madrid

Ángel José Mancha Núñez
Ruth Navarro Carrasco

May 2022

Group 88

Team 5

1. Equivalent Classes

- **First function**

We have considered that the equivalent classes method for defining the different tests was the most suitable method for the first function where a date in iso format is introduced as a parameter to the function “get_vaccine”.

Therefore the equivalence classes and boundary values that we have taken into account are the following ones.

Since we were checking the “date” parameter with a value equal to “2022-03-18” for example, and knowing that the current date of the system is “2022-03-08”, the boundary values that could be defined were a date just below the range of possible values (“2022-03-07”, therefore invalid date) and a date within the range of values (“2022-04-27”, valid date). For this particular example, there is no upper bound so we cannot consider a boundary value just above the range of possible values.

For the equivalent classes, we have considered examples that contemplate values for the date which do not comply with the iso format for those invalid tests, and also examples for the valid tests.

- **Second function**

In addition, equivalent classes and boundary values have been considered for the keys belonging to the json file introduced as input to the second function where the appointment needs to be canceled. In this way, we check the length of the date_signature, the correctness of the cancelation type (either Final or Temporal), and the length of the reason which can not exceed the limit of 100 characters.

2. Grammars and derivation trees

Taking into account that for the second function where an appointment must be canceled either definitely or temporarily and a json file is needed to store the requirements of the cancelation, we thought it would be a good approach to implement the syntax analysis method in order to develop the tests.

The json file had the following format:

```
{
    "date_signature":"<String having 64 hexadecimal characters>",
    "cancelation_type": “Temporal | Final”,
    "reason": "<String with 2 - 100 characters>"
}
```

As appreciated in the image, the json file is composed of a dictionary with three keys. The first key contains the date_signature of the appointment willing to cancel, the second one specifies whether the cancellation needs to be final or temporal for future updates, and the third key contains the reason for the cancellation.

Taking into account the mentioned information, we have developed the following grammar.

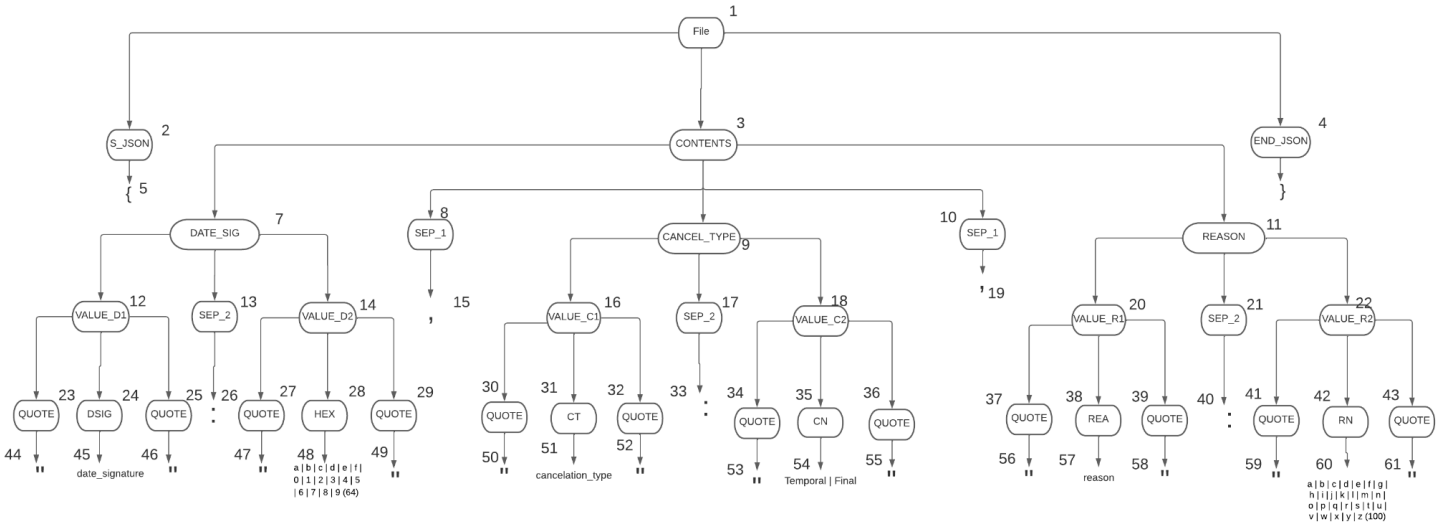
S::= S_JSON CONTENTS END_JSON
S_JSON::= {
END_JSON::= }
CONTENTS:: = DATE_SIG SEP_1 CANCEL_TYPE SEP_1 REASON

SEP_1::= ,
DATE_SIG::= VALUE_D1 SEP_2 VALUE_D2
SEP_2::= :
VALUE_D1::= QUOTE DSIG QUOTE
QUOTE::= “
DSIG::= date_signature
VALUE_D2::= QUOTE HEX QUOTE
HEX::= a | b | c | d | e | f | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 (64)

CANCEL_TYPE::= VALUE_C1 SEP_2 VALUE_C2
VALUE_C1::= QUOTE CT QUOTE
CT::= cancelation_type
VALUE_C2::= QUOTE CN QUOTE
CN::= Temporal | Final

REASON::= VALUE_R1 SEP_2 VALUE_R2
VALUE_R1::= QUOTE REA QUOTE
REA::= reason
VALUE_R2::= QUOTE RN QUOTE
RN::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z (100)

Here we provide the derivation tree



We have proceeded with this method by duplicating and deleting the non-terminal nodes and modifying too only the terminal nodes up until the most relevant cases have been considered.

3. Methodology

For the second function, even though it is not asked, we consider it valuable to make some clarifications regarding some of the approaches that we have followed to better understand the code.

Regarding the cancelation type (Final or Temporal), we have decided to write the date signature of the vaccination appointment that needs to be canceled and the cancellation reason into a new file named *store_cancelation.json* that will contain the information of the appointments for further statistics. Furthermore, if it is final, we erase the data from the json related to that appointment with the corresponding date signature.

On the other hand, if the cancellation type is Temporal, the store_date.json has been modified and a new label has been considered to mark that appointment as canceled, but under no circumstance, the appointment will be deleted permanently, just so that it can be reactivated in the future.