Backend use cases - configurations

Configurations Use cases backend :

1) Add new home = add a certain number of new sensors
   - Constraint : we base all the modifications on the excel file : every time we modify something, we update the file, and generate a new configuration in Cassandra.
   - Behaviour :
     - It creates a new configuration = new instance of the excel configuration file (input)
     - We detect that those sensors doesn't have any missing data nor data in raw table
     - We query data since their initial timestamps
     - Then, new data is added to cassandra tables (output)
   - Priority : High
2) Remove a sensor from a home
   - Behaviour :
     - Data from this sensor are still in the database, as well as in tmpo, but the sensor is not in the configuration anymore (input)
     - The power data and groups data for the home without this sensor will simply change based on the remaining sensors since the sensor removal time (output)
     - Since the removed sensor is not definitely removed from tmpo, we can still get back missing data from the previous configurations (containing the newly removed sensor)
   - Priority : Medium : it can simulate a failure of the sensor
3) Remove a home
   - Behaviour :
     - Idem as 2) = remove a set of sensors : generates a new configuration without this home (input)
     - If the home belonged to a group, it is removed from it (we get 5))
     - with the new configuration, we simply do not take this home into account
     - Previous data is still in cassandra, but stop when it is removed from the config (output)
   - Priority : Low : if a neighbour wants to stop the experiments for instance
4) Add home to a group
   - Behaviour :
     - Generates a new configuration file with new group
     - In the database, it will change the data since the addition time according to the new group composition (output)
       - If the new home was already in the previous config, then the data for this group will start
   - Priority : Medium : If it happens, it is either from an existing home from another group, or an existing home that belongs to 0 groups.
5) Remove home from a group
   - Behaviour : Same as 4)
6) Create new group
   - Behaviour :
     - It creates a new configuration with a new group (either with new homes or with existing homes. Either way, it will just create a new group) (input)
     - The new group data stored in the database will start at the time the new configuration is considered, which means that all data before this time will not be in the database (output)
       - (except if we use recomputePower to get data since the beginning of the group which is defined by the first raw data timestamp amongst all homes)
   - Priority : High : very likely
7) Invert two sensors :
   - for instance if we move a flukso from 1 home to another : data from a sensor will not belong to

the same house
        - Constraint : Need to specify in the excel the date of inversion, so that we put a missing data with that date. Otherwise, the system will just continue the data from the last registered timestamp instead of recovering data since the sensor's initial timestamp (in his new home).
    - Other case : invert 2 sensors from the same home : same impact, we need to recompute all data for both
    - Behaviour :
        - new configuration with inverted sensors
        - With the given dates in the excel file, get data since initial timestamp for those sensors (output)
    - Priority : quite low : it is possible
8) A participant quit, and ask to remove his sensors from the system
    - Constraint : Sensor not in tmpo database anymore : we cannot query it without getting an error
        - Cannot use recomputePower on this sensor anymore
    - Behaviour : It will negativelly impact the execution of the queries (crash) : only if there are missing data from the removed sensor : the sensor will try to get those data back using previous configuration containing this sensor, but it is not in tmpo anymore.
    - Priority : low : it is very unlikely to happen