Backend use cases- configurations

Configurations Use cases backend :

1. Add new Flukso (set of sensors) to a home
   - 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 the sensors doesn't have any missing data nor data in raw table
     - We query data since its initial timestamp
     - Then, new raw data is added to cassandra tables (output)
     - The other sensors of the home behave as usual : recover potential missing data + get new data
     - For the *power* data : it is recomputed from the earliest timestamp amongst all sensors of the home (here since the initial timestamp of the newly added sensor)
   - Priority : Medium

2. Add new home = add a certain number of new sensors
   - Constraints :
     - 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.
     - A sensor belongs exclusively to 1 home.
   - Behaviour :
     - Either all the sensors of the new home are new, or some were already used and data are already present in the db
     - 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

3. Remove a sensor from a home
   - Constraints : same as 1)
   - 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

4. Remove a home
   - Constraints : same as 1)
   - 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

5. Add home to a group
   - Constraints : same as 1)
   - 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.

6. Invert two sensors :
   - Constraints :
     - same as 1)
   - Behaviour :
     a. If we move a Flukso from a home to another : the sensors id remain the same, but the data needs to be adapted to the new home (previous data is kept in db).
        1. Solution 1 : Specify in the excel the date of inversion, so that we put a missing data with that date. Otherwise, the system will just continue to store data from the last registered timestamp instead of recovering data since the sensor's initial timestamp (in his new home).
        2. Solution 2 : Or using the insertion time and the sensor id to create a unique id per

home. (simpler)

b. Other case : invert 2 sensors from the same home : same impact, we need to recompute all data for both

- new configuration with inverted sensors generated
- Get data since initial timestamp for those sensors (output)

- Priority : very low : it is possible but very unlikely

7. A flukso crashes :
   - Constraint : manually change a constant
   - Behaviour :
     - If the flukso is recovered within a certain amount of time (< the maximal missing data recovery time), then the backend will eventually get back missing data due to the crash. Otherwise, data will be lost from the time of the crash until the time missing data can be recovered.
       - *Solution* : change the constant "LIMIT_MISSING_RAW" to fit the time needed to recover all the data.
       - Todo : automatically make a signal when a flukso crashes with a certain delay with respect to the limit of missing data.
   - Priority : low : It can happen

8. A participant asks 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 negatively 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.
       - *Solution* : try catch : before creating the tmpo session, check every single sensor's validity.
   - Priority : low : it is very unlikely to happen