* Controller
  + Id/username of device/user
  + Connected devices
    - Show if user is in the house/system or not
  + Give status overview of all lights to user
  + Change state of specific light
  + All users registered
    - Enter or leave house -> information message to all users in the house
    - Last user leaves house -> lights turned off
      * First user enters house -> last known light state is used
  + Show inventory of specific fridge
  + Inform all clients fridge is empty
  + Open fridge requested by user
    - All communication between fridge and user are done without controller
  + Save most x recent temp measurements (configurable by client)
  + Show current temp in house (average if multiple sensors in place)
  + Show history of temps (average if multiple sensors in place)
* User
  + Connect to controller and receive unique id/name
  + Enter or exit system **(leaving or entering house?)**
  + 1st connection -> user in house
  + Ask list of all connected devices/users
  + Ask state of all lights
  + Change state of specific light
  + Ask inventory of specific fridge
  + Ask to open fridge
    - Remove/add items to fridge
    - Close fridge when done
    - No other connection possible when user is using fridge!!!!
  + Ask current temp
  + Ask history of temp
* Temp sensor
  + Connect to controller and receive unique id/name
  + When sensor is launched, set begin temp
  + Every x seconds -> send new measurement
    - X being practical value
    - Temp calculated by: adding random value between -1.0 and 1.0 degrees to previous value
* Fridge
  + Connect to the controller and receive unieque id/name
  + Have list of inventory **(history?)**
  + Return list of its inventory
  + Add or remove item to inventory
    - If last item removed -> contact controller -> inform all users fridge is empty
    - Only possible if fridge has been opened
    - Only the user that opened can add or remove item
* Light
  + Connect to controller and receive unique id/name
  + Change current state (on/off)
  + Return current state