1. Data from the sensors are sent to the server on sensor change, periodically or both? (for example, depending on the type of sensor).

Sensors report data to server only if there are changes of its data by socket.(To reduce the pressure of server and usage of network)

1. Data from sensors are sent one by one or all together, e.g. triggered when the button is pushed. If data is sent only when the button is pressed, then if no one presses the button in the classroom how will light and presence sensors be sent to the server (this information might be needed when the Web or Mobile APP are used)?

Data from sensors are sent one by one to server when data are changed. And then preserved in the server.

Plus, we considered button is kind of sensor which send triggering message to server.

3. When a button is pressed in a room or when the state of any other sensor changes, that information must be sent to the server (web services):

* What if the format of the messages? JSON?
* Are RESTful webservices being used? Have you already thought about the programming language and the server type (e.g. Apache)?
* How does the server “know” to which room that sensor belongs to? Is it sent any identifier? If so, that identifier is the identifier of the sensor or the room?
* The server (web services) will send that information only to the Inteligent Controller module, only to the Database module or to both modules?
* If you store sensor data in the database, then will you store only the last value of each sensor or all values?

For server part:

* All the message which sent in or out are in JSON format.
* We provide RESTful API for client to access. We use Python’s Flask to support all services.
* Each hardware must have a unique ID when it was manufactured. And each room has its own unique ID. We get the relations between hardware and rooms from DB.
* Server gather necessary information and then only send them to IC module.
* Server store non-persistence data like sensors’ data and we only keep their last available values.

4. Regarding to the Inteligent Controller module:

* It receives data from multiple rooms, therefore from multiple sensors, so will it keep all that information in RAM or will retrieve it from in the database only when it is needed?
* Because we have multiple rooms, how does the Inteligent Controller module know to which room a sensor belongs?
* What about the result of the light control will it be kept in RAM or sent to the Database? The option taken will have implication on how the light controller gets the light state information?

5. If instead of just a few rooms you have one million rooms, do you need to modify anything? For example, will the RAM of the server be enough? Are you thinking about the most effective data structures to store data in RAM? Should you have all data in RAM all the time? Are the database tables ready for such a number of systems?

For current version, server save non-persistence data in form of Key-Value in RAM. If we have more rooms, we would use a built-in Redis to save these data.

But if we have one million rooms, I believed that a complex distributed system architecture are needed. And the performance of this kind of system cannot be judge simply by observing, prototypes and tests are needed.

6. How does the light “know” that must be turned ON or OFF? Are you considering the following options?:

* The server sends a message to the raspberryPi? If the answer is yes:
  + How will it work?:
    - Will you have a web service on the light controller to receive commands from the server? (If so, how do you deal with private IP addresses?)
    - You will keep a socket open? If, how do you deal with the number of ports that need to be keep open and private IP addresses?
  + How does the server know to which raspberryPi send the light ON/OFF command?
* The server sends back to the client the light state, as a response, when the state of the light switch is sent to the server by the hardware module. If the answer is yes:
  + What happens when the command to turn ON the light comes from the web application or the Android APP?
  + And what if the light is not in the same raspberryPi as the sensors?
* It polls the server asking for the light state? If the answer is yes:
  + The server “asks” the state to the Inteligent Controller module or the Database?
  + How does the sever “know” which light is asking for the state?
  + What is the interval between requests?

Every hardware(sensor / light) will build socket connect with server when they are online even if two hardware running on the same Raspi(If failed, they will keep trying to connect to server). And Server will mark this socket with its unique ID. Server send command to hardware by socket. By using this technology, server must have one IP which one single port opened to the public.

Server will not send the result of command to client. Server only tell client that the command is sent or the command is invalid. The client will update light’s state from server shortly after sending the command. (We don’t care which raspi the sensor is running on.)

Server will store non-persistence data like light’s state and sensor’s data. So when queries are coming, server will simply make the response. All hardware are mark with unique ID to identify who they are. The interval is decide by the client.

7. One raspberryPi can have only sensors, only lights or both. How do you relate this in the database?

We consider each single sensor and light to be a hardware program. We don’t care about which hardware the program is running on.

1. On the server will you be using the same programming language for all components? If not, how do you plan to do the integration? For example, the Webservice must “talk” with the database module. By the way, will you allow direct access to the database, or will you build an API to access it (e.g. using a Data Access Object – DAO).

Each part will develop their own part independently. We will use a series of built-in API, but this is under discussion.

1. In the raspberryPi will you be using the same programming language for all components? If not, how do you plan to do the integration?

Yes, by python, but this is under discussion.

1. There are two groups working with hardware modules. One with the sensors and the other with the lights. These groups have one module in common, the communication module. Which features of the module will be common for both groups? Which will not?
2. For now, the permissions of Administrators and Teachers are the same, but is the system ready for, in the future, add more permissions to the Administrator?
3. You are using authentication for the users. What about for the raspberry Pi modules? How do you know if you can trust them?

Each hardware has its own unique ID. If hardware’s UID is registered in database and it provide correct authentic key when build connections, then we will trust it.

13. When a user, is using the mobile application or the web application:

* If the user forces the light to turn ON or OFF, what happens? Is the Inteligent Controller that decides if the light is turned ON or OFF? Is the database module?
* When the user sees the light status or any sensor status, that status is retrieved from the database or from the Inteligent Controller module?
* The logic in Intelligence Controller can be changed. So this is the problem for user.
* Info is retrieved from server.

14. Can you identify all the uses cases, from source to target, that are related with the following operations (only the first examples have the complete flow):

1. New data from a sensor is available, it is sent to the communications module, that sends it to the server, in the server data is sent to the Intelligent Controller and Stored in the database (if applicable);

User Case 2.4.2

1. The Intelligent controller has an output to be sent to the light controller (and stores in database, stores in RAM or sends to the light controller – whatever applicable in your design);

User Case 2.4.5

1. Output of the Intelligent Controller reaches the light controller and the light is turned ON or OFF (e.g. the light controller module calls the Communicatons module, a request is sent to the Webservice, data is requested to the Intelligent Controller or the database, sent back to the calling Communications module, returned to the caller controller module, that turns the Light ON or OFF);

User Case 2.7.2

1. The user presses, in the Web or Android Application, the button to turn ON or OFF the light. Similar to i), but the user permissions and “priority” level must be checked;

User Case 2.3.3

1. User logs in the application (Web and Mobile);

User Case 2.3.1

1. User checks the sensor status (Web and Mobile Applications);

User Case 2.3.2

1. Other features that where considered in the SRS.

Emmm... Plus, what is the communication module?

**(Some) Questions related to the contents of the Software Requirements Specification**

1. I use Ubuntu Linux 18.04, so I cannot use the system?

Yes and no. We cannot guarantee this software can run stably on Linux 18.04. If the customer strongly wants to run this software stably on Linux 18.04, we can discuss this problem later.

1. Communications between sensor and server is made using plain sockets or web services?

Socket

17. Are the Android APP and the Web APP use cases the same?

18. Was the dataflow between components planned? Some examples:

* 2.3.1 “User Login”, that “connects” to 2.3.2 “Verify Login” that will have as a consequence 2.6.4 “Sever Wants Authentication of the User ID and Password”. In this case we have a flow a data from the source “The User” to the sink “The Database”. OK!
* What about when a sensor changes state or a light needs to be lit? The use cases start in the server only (2.3.7(2.7.2))?

19. You must assume that the sensors can be in different raspberryPi, so what is the impact in 2.37(2.7.2)?