Use-Case Specifications:

1 Data visualization

1.1 General Information

1.1.1 Use-case description

To analyze and share the data it's necessary to have some kind of output that is readable by humans and computers. To achieve that some kind of visualization is necessary. A download of the data should also be available in a standardized format.

1.2 Scenarios

1.2.1 Main Scenario

1. The user selects some filter options that are used on the available data. *(a - filter data)*

2. If the data is older than 24 hours, a process is invoked to fetch the latest available data from the IoT-Boxes. *(b - update/collect sensor data)*

3. Depending on the data different kinds of charts are shown representing it. Possible charts to show changes over time would be the line chart, historical bar chart and stacked area chart. To show the range of values (i. e. safe values for gas or UV light) a box plot chart is used.

4. The user downloads the data that is visualized right now and exports it as a csv file.

1.2.2 Alternative Scenario

IoT-Box is offline

2a. Not all boxes can be reached. The user will be informed that the data is incomplete and which IoT-Boxes are offline (error code 101 - IoT-Box unavailable).

No data available

3a. There is no data that can be shown as of now because no data was generated yet or it couldn't fetch any. The user is informed about that situation.

2 Filter data

2.1 General Information

2.1.1 Use-case description

It's possible to filter data based on the user's intent because not every available data value is necessary for him/her. Different filters allow a detailed overview with just a snippet of the accumulated data.

2.2 Scenarios

2.2.1 Main Scenario

Extends 'data visualization'

1. First the user selects what boxes will be shown or hidden in the final result. He can choose between one IoT-Box or all of them.
2. The user selects what kind of sensor data he wants to have included. He/she can choose between temperature, humidity, pressure, VOC, eCO2, ultraviolet, infrared and visible light. One or more need to be chosen.
3. Finally the user selects the dates that are important to him. Sensor data recorded between two dates (including themselves) are filtered. It's also possible to select just one day.

2.3 Postconditions

2.3.1 Filtered dataset

After the user selected the options to filter the data it gets visualized and a download is available (use case 'data visualization').

3 Register testbed

3.1 General Information

3.1.1 Use-case description

To get sensor data it is necessary to connect the IoT-Boxes to the server. By registering them to the server it's possible to update the data when necessary.

3.2 Preconditions

3.2.1 Precondition one

The user needs to be logged in as an admin to update this information.

3.3 Scenarios

3.3.1 Main Scenario

1. After the user logged in he's/she's able to put in the IP-address of the IoT-Box.
2. He also needs to create a cluster by putting in a name or choose a previously existing one.
3. The IP-address is pinged and added to that cluster if it's a 'UniVie IoT-Box'.
4. The user can assign a name to the IoT-Box to recognize it more easily if the cluster gets bigger.
5. The data will be updated for the first time and added to the existing data set of the other boxes (use case 'update/collect sensor data')

3.3.2 Alternative Scenario

IP-address not reachable

2a. The IP-address can't be reached or the device behind it is not recognized as an 'UniVie IoT-Box'. The user is asked to put in a different IP-address.

3.4 Postconditions

3.4.1 New IoT-Device

A new IoT-Device is added to the cluster and will be included in the updates during the use case 'update/collect sensor data'.

4 Delete testbed

4.1 General Information

4.1.1 Use-case description

It's also possible to delete a previously registered IoT-Box and testbed from the cluster. It won't be included in data updates after it has been removed.

4.2 Preconditions

4.2.1 Precondition one

At least one testbed needs to have been added before to the cluster.

4.3 Scenarios

4.3.1 Main Scenario

1. The user selects one or several of the available IoT-Boxes to delete.
2. Sensor data will be pulled one last time before it's removed to make sure that no data is missing (use case 'update/collect sensor data').
3. It is removed from the cluster and the user gets notified about the successful removal.

4.4 Postconditions

4.4.1 Testbed deleted

The IoT-Box and testbed is deleted from the cluster and won't be included in any further data updates until it's added again.

5 Update Code

5.1 General Information

5.1.1 Use-case description

To adapt to different use cases of the boxes it's possible to update certain parts of the processing code. The user can write and push parts of codes to the boxes over the internet.

5.2 Preconditions

5.2.1 Precondition one

The user needs to be logged in as an admin to update code.

5.3 Scenarios

5.3.1 Main Scenario

1. The user selects a cluster that he wants to update.
2. The code will be put in by the user. To help him/her as much as possible it would be good to have an auto-complete editor with highlighting.
3. The code will be checked if there are any syntax errors.
4. The user sends the code to the cluster of IoT-Boxes if no syntax errors exist.
5. The code will be run on the devices (use case 'run code').

5.3.2 Alternative Scenario

No cluster

1a. The user can't continue if there is no cluster yet that can receive an update.

Syntax errors

2a. The user can't continue if there is no cluster yet that can receive an update.

Runtime errors

5a. The user will be informed about any runtime errors that occur, like running code for sensors that aren't connected, unresponsive boxes, endless loops.

5.4 Postconditions

5.4.1 Updated code

The user code in the selected cluster of boxes is updated and will be run during the use case 'run user code'.

6 Update/collect sensor data

6.1 General Information

6.1.1 Use-case description

It's necessary to collect sensor data from the IoT-Boxes and update them on the server to visualize and export it.

6.2 Preconditions

6.2.1 Precondition one

At least one IoT-Box needs to have been added before to the cluster.

6.2.2 Precondition two

Some user code must have been pushed to the IoT-Boxes successfully.

6.3 Scenarios

6.3.1 Main Scenario

1. All the registered IoT-Boxes are being iterated to fetch data they have been collecting.
2. The IoT-Boxes send their data back to the server where it's being aggregated and stored.
3. A message will be send back to the IoT-Box to notify them about the successful delivery of the data.

6.3.2 Alternative Scenario

IoT-Box offline

1a. If an IoT-Box is offline then no data can be send back. Another attempt will be started before an error code (101 - IoT-Box unavailable) will be displayed.

Data not received

3a. Another attempt will be started (2.) if no notification comes back. A log will be saved with date and error code (102 - Request Timeout) if there's still no message about the successful delivery.

6.4 Postconditions

6.4.1 Sensor data received

The Requester receives a copy of the latest sensor data from each of the IoT-Boxes.

7 Read sensor data

7.1 General Information

7.1.1 Use-case description

To analyze, visualize and export data it's important to collect it first. The IoT-Boxes read the sensor data over the connection through I2C.

7.2 Preconditions

7.2.1 Precondition one

Sensors need to be connected to the Raspberry Pi IoT-Boxes

7.3 Scenarios

7.3.1 Main Scenario

1. Depending on the user code (use case 'run code') the connected sensors data is read.
2. New incoming sensor data is constantly saved in a file (one per month) that will be send out if needed (use case 'update/collect sensor data')
3. Sensor data that is older than a year will be deleted from the IoT-Box to preserve some disk space.

7.3.2 Alternative Scenario

Sensor not connected

1a. If the data can't be read because the sensor is not correctly connected a message will be send to the server to inform the user that the code isn't executable (error code 103 - Unprocessable code).

No disk space left

2a. No new data can be recorded because there is no free disk space available anymore. The data collection will be stopped.

7.4 Postconditions

7.4.1 Sensor data recorded

A file is created with the collected sensor data ready to be send out if needed.