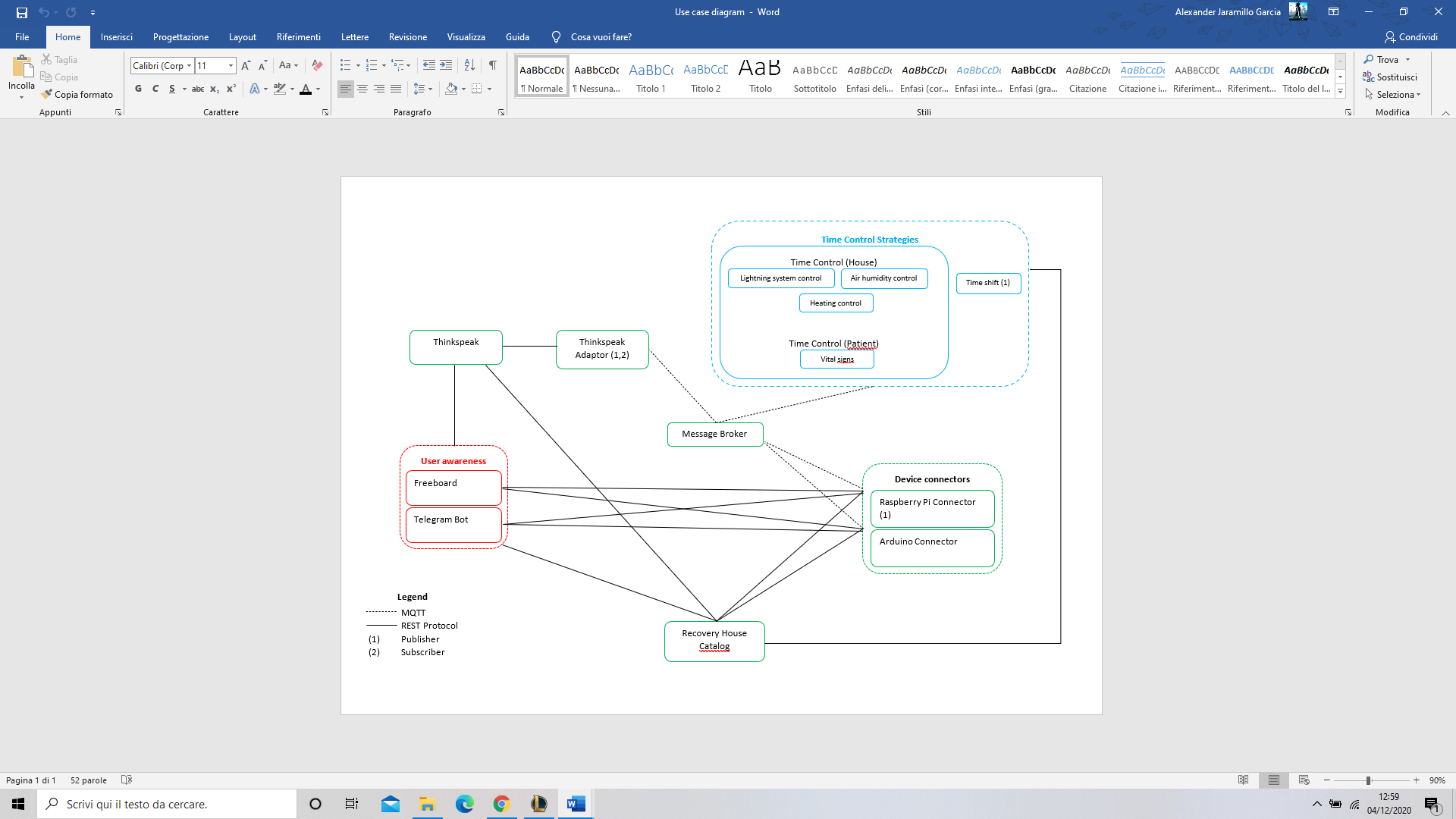
1. Name of Use Case

|  |  |
| --- | --- |
| **Name of the Use Case** | **IoT Platform for a Smart Recovery House** |
| **Version No.** | v0.1 |
| **Submission Date** | 09/12/2020 |
| **Team Members (with student ids)** | JARAMILLO GARCIA Alexander s227353  FARAMINI Elahe s289579  BALAN Rares Alexandru s290175  MANHAEVE Alexander s287432 |

1. Scope and Objectives of Function

|  |  |
| --- | --- |
| **Scope and Objectives of Use Case** | |
| **Scope** | The proposed IoT platform aims at providing services for a smart recovery (for patients with no hospitalization needed) house management |
| **Objective(s)** | The expected results consist on providing a smart control of appliances to set a suitable and friendly environment for a patient in state of recovery |
| **Domain(s)** | Smart House, Smart Environment, Smart building, Smart Health-Care, Smart Medicine |
| **Stakeholder(s)** | Citizens, Doctors, Medicine, Nursery, Medical research. |
| **Short description** | The proposed IoT platform aims at making smart a house to facilitate and improve the recovery phase of a patient with No-High-level illness. It integrates different IoT devices for managing appliances in a common house environment. It provides control strategies for environment and body conditions.  The overall platform provides unified interfaces (through both REST and MQTT) to integrate the house into Smart Building and Smart Health-care environments. Hence, Demand/Response rules can be applied. The platform provides end users with detailed knowledge of the vital measurements and the actual state of all the devices trough time.  The main features are:  • Remote control of some appliances  • Control strategies for lighting, heating, aerial humidification and vital signs as heart rate, oxygen saturation, body temperature, sleep cycle tracking  • Manage the resources at hand to enable Demand/Response  • End-user applications for house environment and patient control |

1. Diagram of Use Case



1. Complete description of the system

The proposed IoT platform for Smart Recovery House follows the microservices designing pattern. It also exploits two communication paradigms:

1. publish/subscribe based on MQTT protocol.
2. request/response based on REST Web Services.

In this context, ten actors have been identified and introduced in the following:

* The **Message Broker** provides an asynchronous communication based on the publish/subscribe

approach. It exploits the MQTT protocol.

* The **Recovery** **House Catalog** works as a catalog for all the actors in the system. It provides information about

end-points (i.e. REST Web Services and MQTT topics) of all the devices, resources and services in the platform. It also provides configuration settings for applications and control strategies (e.g.

timers, list of sensors and actuators). Each actor, during its start-up, must retrieve such information

from the Home Catalog exploiting its REST Web Services.

* The **Raspberry Pi Connector** is an implementation of the *Device Connector* that integrates into the

platform raspberry pi boards. Each raspberry is equipped with temperature, humidity, and light intensity sensors to provide environmental information about the status of the house. It provides Rest WebServices to retrieve environmental information (i.e. environmental temperature, humidity, light intensity inside the house/room). It also works as an MQTT publisher sending information on the same measurements.

* The **Arduino Pi Connector** is an implementation of the *Device Connector* that integrates into the

platform Arduino boards. The Arduino is equipped with relays and electro valves to switch on and switch off the connected appliances (house lighting system, heating system and air humidifier). It provides Rest Web Services to retrieve and change the status of the

connected appliances (on/off/intensity). It also works as an MQTT subscriber to receive actuation commands from other actors that exploit the MQTT protocol (e.g. Control Strategies).

* The **Timer Control** is a control strategy that manages all the appliances described in the previous point. The strategy consists on switching on certain actuator when the related measured value overcomes a certain threshold value (that can be set from user interface). Each area (room) of the house is managed by an instance of this strategy.

It works i) as an MQTT subscriber to receive information on the measured parameters; ii) as an MQTT publisher to send actuation commands to IoT Devices (Arduino).

* The **Time Shift** is a control strategy to manage appliances depending on time-schedules provided

by the **Recovery House Catalog**. In our case, it allows users to switch on/off the light systems from 6:00 to 21:00 by default or setting a new time control depending on what the stake holder wants. It works as an MQTT publisher to send actuation commands to IoT Devices.

* The **Thinkspeak Adaptor** is an MQTT subscriber that receives measurements and upload them on **Thinkspeak** through REST Web Services. It works also as an MQTT publisher to send command signals (that they have been calculated by processing the collected measurements into Thinkspeak with Matlab code) to Arduino (that will provide the actuation).
* **Thinkspeak** is a third-party software that provides REST Web Services. It is an open-data platform for the Internet of Things to store, post-process and visualize data through plots. Its working details has just been explained.
* **Freeboad** is a dashboard to retrieve data from IoT devices and visualize them exploiting the REST

Web Services provided by **Raspberry Pi** and **Arduino Connectors**. It also exploits the **Thinkspeak**

Web Services to import plots about collected measurements.

* **Telegram Bot** is a service to integrate the proposed infrastructure into Telegram platform, which is

Cloud-based instant messaging infrastructure. It retrieves measurements from IoT devices exploiting

the REST Web Services provided by **Raspberry Pi** and **Arduino Connectors**. It also allows users

on sending actuation commands to IoT devices again exploiting REST.

1. Desired Hardware

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |