Motivation

- Our motivation for this project was to tackle a real life issue. When we were discussing for project idea, we realised that HOAS Houses had the issue of not knowing if your roommate is present in their room or not.
- We wanted to solve this issue by creating a web application system, where you can check if your roommates are in their room or not. While creating the system we decided to add extra features such as; ability to see the rooms temperature, a “don’t disturb” button, to notify your roommates that you don’t want to be disturbed.
- Originally the project also had the feature to see how many persons are in the house itself. This is a feature we thought that can be used by firefighters in case of a fire, to check how many persons are possibly inside. But we didn’t put that future to the web application because the presentation we did was a prototype, but it is possible to implement that future quiet easily.

Connection To Literature

- Then we started brainstorming on ideas how to create our idea. We understood that we need a microprocessor, that can connect to wifi, and has enough GPIO PIN’s for us to do prototyping with breadboard. Of course we knew that we will need motion sensors, which we decided on using PIR Motion sensors, because they were cheap. A simple temperature sensor for measuring temperature, a button for the don’t disturb feature, LEDs to display the situation of the room such as if somebody is inside the green light turns on, and other technical components such as cables, breadboard etc.

- **ESP8266 Node MCU v3** is the microprocessor used in this project. Main reason why we used this model in particular was, that it is 6€ making it cheap . Furthermore we needed at least 7 GPIO pins for the 2 PIR sensors, 2 LEDs and the temperature sensor. Another good thing is, that it has built-in WiFi which we are using to send the data. It’s main job is to receive the data from the sensors, and send them to the Node.js server. Beside that, the ESP8266 fetching the current time from an NTP server. We used this for the timestamp we add to every request. We are aware that we could have put the timestamps in the database too. However, we just wanted to try this on the ESP. Also we use the MAC address of the device to differentiate the rooms. Depending on the MAC address we can understand which data belongs to which room.

- **PIR Motion sensors**, are the key to our project. To understand an entry or an exit through a door to the room, we used **two** PIR motion sensors. This was key to our project, because using two PIR motion sensors enabled us to understand if a person entered a room, or exited a room.

- We can understand how these two PIR motion sensors are placed so that they can be useful to the project by picturing a door of a room. So there is the side of the door, that you see when you enter the room. We will call this side “A” to reference it later. Then there is the side of the door that you see when you exit a room, we will call this side “B” again to reference it later.
- So imagine we place one sensor for each side, side “A” and “B”. If the motion sensor on the “A” side has been triggered first, and the motion sensor in “B” side has been triggered second, we can understand that there has been entry to the room. This is how we differentiate if someone enters or exits a room, by which motion sensor has been triggered first. As another example, if motion sensor in the side “B” has been triggered first and the motion sensor in the side “A” has been triggered second, we can understand that a person has exited the room. This would have been impossible to detect with only one motion sensor.

- **Temperature Sensor** is only used in a way to measure the temperature and display it on the web application but our original intention was that it can be useful in situations such as fire hazard.

- **LEDs** are used to show the state of the room. So if there are no lights turned on that meant that nobody is in the room. If the light is green it means that somebody is inside the room, and finally if the red light is on, it means that the person inside the room doesn’t want to be disturbed.

- A simple **button** is used that you can press it and it would change the state of the room, states being “Don’t disturb” and “Available”.

- **Node.js Server** is one of our backend. It is used to process the data coming from the microprocessor (ESP2866). ESP2866 sends a JSON Object file containing data to the Node.Js server. The server routes the different requests and persist everything into the right tables in the MYSQL database. For simplicity we used the Express Framework. With this framework it is possible to write really structured Node.js applications. We decided to use two different routes, one in the case a motion was detected (/motion route) and another one for the do not disturb feature (/dnd route). The *motion route* getting the following POST parameters from the request: deviceId, motionValue, temperature and timestamp. Using this information application builds a SQL UPDATE query to store everything in the database. For the dnd functionality it’s almost the same. We are getting four POST parameters: deviceID, temperature, timestamp and dnd. The dnd parameter is an integer and could be -1, 0 or 1. “-1” parameter means the person doesn’t want to be disturbed. “0” parameter means, that nobody is in the room. “1” parameter means, the person is available.

- **Mysql Database** is used to store data. This project uses relational database and MYSQL, because more than the insertion of data, project primarily updates data which makes relational database more suitable. Web Application uses the username, and the password in the web application to access the data from MYSQL Database, and also uses the MAC address received from the ESP8266 to differentiate rooms.

MYSQL Database has three tables, which are Flats, Rooms and Users. Flats table have the information about the house, such as their description, and there ID. Rooms table have information of; the room owner, the flat that the room is connected to, people inside, MAC address of the ESP8266 connected to the room, availability of the room and the temperature of the room. Users table have information about the room owners name, their password and the room they are connected to.

- **Web Application** is an important part of our project. Technologies used in the web application are HTML, CSS, Javascript, JQuery, and Bulma CSS Framework for frontend and PHP for the backend. Application is started by logging in to the system assigned by an admin. Then when you login in to the page, you can see the status of your room. How many persons are inside the room, the temperature of the room, and the room's status being, "available" or "don't disturb". These informations are all available on the web application. Of course the key feature of the web application is that you can see your roommates information such as wherever or not that they are inside their room, their room's temperature, and if they want to be disturbed or not.

Technical Description Of Sensors

PIR Motion Sensor Module

- PIR Motion Sensor or Pyroelectric Infrared Sensors are basically made of a pyroelectric sensor, which generates energy when exposed to heat. It detects movement of a human or an animal by the heat energy emitted in a form of infrared radiation.
- The module also consists of a Fresnel Lens, which focuses infrared signals onto the pyroelectric sensor.
- The module has just three pins: A Ground and VCC pin for powering the module and an digital output pin which gives high logic level if an object is detected.
- The module has a built in analog to digital converter and also an amplifier, to strengthen the output signal.
- It has two potentiometers. One for adjusting the sensitivity of the sensor and the other for adjusting the time the output signal stays high when object is detected.
- The module has three more pins with a jumper between two of them. These pins are for selecting the trigger modes.
- The first one is called "non-repeatable trigger" and works in a way that when the sensor output is high and the delay time is over, the output will automatically change from high to low level.
- The other mode called "repeatable trigger" will keep the output high all the time until the detected object is present in sensor's range.

BMP-280 Temperature and Pressure Sensor

- **Barometric Pressure is not explained in the documentation because only temperature sensor of the module had been used in this project.**
- Sensor uses a library from sparkfun for reading the data from the sensor.
- Sensor uses I2C Bus protocol to communicate.

During this project we ran into less issues then we expected, but here are the issues that we ran into, and how we dealt with them.

PIR Motion Sensor Issues

- First issue with the PIR motion sensor was one of the major issues for us. We used PIR motion sensors for our project and we needed so that it detects only a single line of movement. Sadly, our PIR Motion Sensor were detecting everything. They were more sensitive than we expected. We fixed this problem by putting the motion sensors inside a toilet paper roll. This wasn't the best solution, but it definitely solved our problem in a big way.
- Another problem was the wait time. For PIR motions to detect new data, they had to wait for 8 seconds. Sadly, this was the only issue we couldn't solve, but still wasn't a major problem when we were presenting the prototype.

Technical Part's Issue

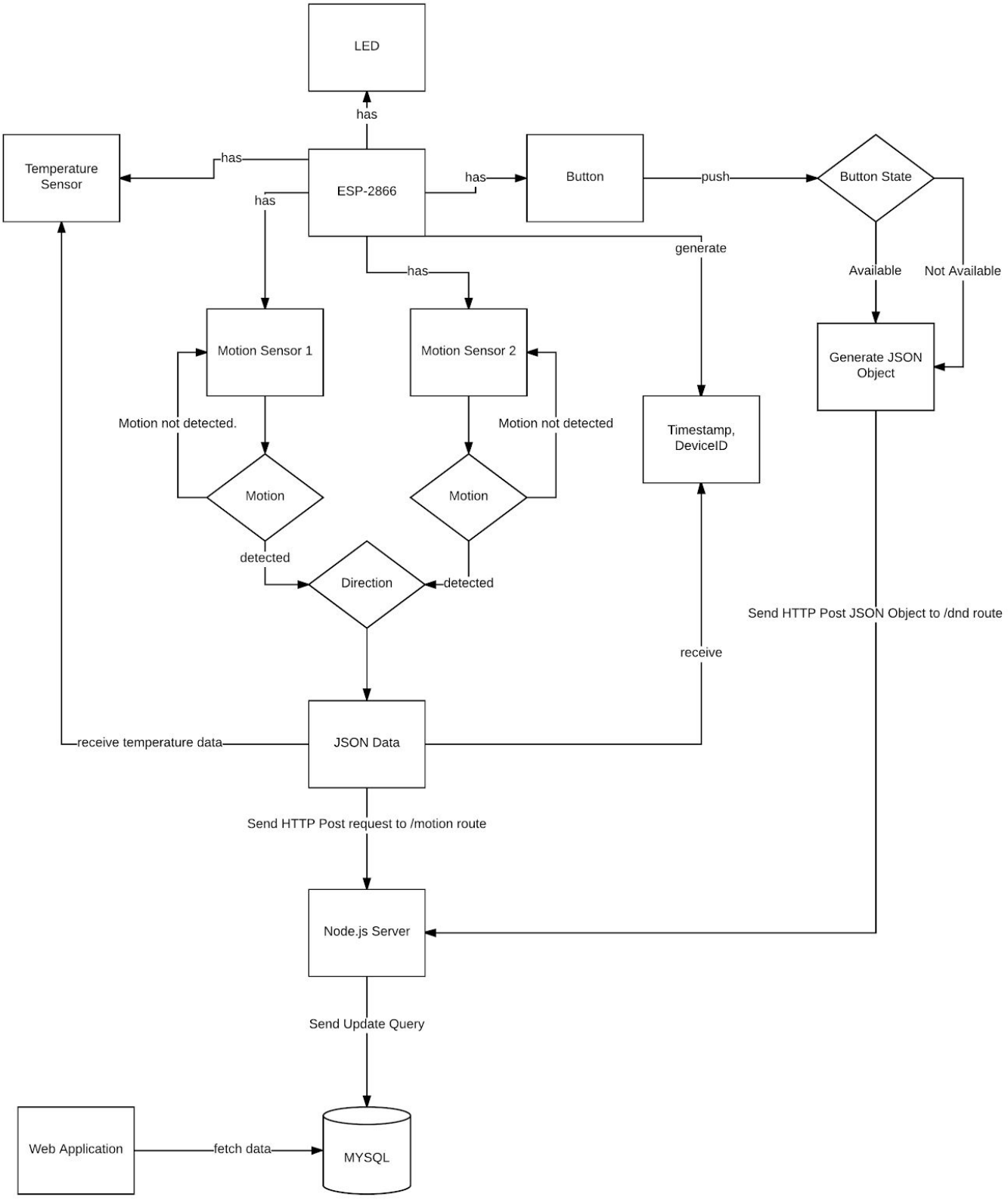
- This was a problem at the beginning of the project that we didn't know which parts to buy because both of us never done a project like this before. So we ordered some wrong parts for our project. Luckily, we managed to do the research and order new ones and they got in time, but we lost a lot time which was valuable to us. We learned from our mistake, and will do better research on our future problems.

Time Issue

- This might seem that it's not a big issue but for us it was. We simply couldn't manage time! With both of us being exchange students and having booked trips before, there were some cases that sometimes one of us wasn't at the city. We separated the work in different parts and each of us was responsible for a dedicated field of duty. In the end we worked more on the days we were both where in the city, or worked remote from distance.

Project Member’s Impact On The Project

Project Component	Turhan Ali Gür	Josua Öhler
Hardware & Sensors	30%	70%
Front End Web Development	70%	30%
Back End Development	40%	60%
Presentation & Documentation	50%	50%
Research	50%	50%


LED
has
Temperature Sensor
has
ESP-2866
has
Button
push
Button State
has
has
generate
Available
Not Available
Motion Sensor 1
Motion Sensor 2
Generate JSON Object
Motion not detected.
Motion not detected
Timestamp, DeviceID
Motion
Motion
detected
Direction
detected
Send HTTP Post JSON Object to /dnd route
receive
receive temperature data
JSON Data
Send HTTP Post request to /motion route
Node.js Server
Send Update Query
Web Application
fetch data
MYSQL