# ILikeTheWayYouMove

- Library Movement tracking

Software Engineering

University of Southern Denmark


**Internet of Things Project, 2nd Semester Master**


| **Authors** | **Exam no.** | **Email** |
|---|---|---|
| Christos Anestis Georgiadis | 450520 | chgeo17@student.sdu.dk |
| Chris Bang Sørensen | 65255670 | chso314@student.sdu.dk |
| Thanusaan Mahalingam Rasiah | 410033 | trasi14@student.sdu.dk |
| Veronika Zemanova | 311071 | vezem17@student.sdu.dk |

| **Supervisors** | **Email** |
|---|---|
| Aslak Johansen | asjo@mmmi.sdu.dk |
| Mikkel Baun Kjærgaard | mbkj@mmmi.sdu.dk |


Project Due: 25. May 2018

# Contents

# 1 Introduction

This section aims to introduce and present the reader to the domain of what the project group has worked on while implementing an IoT project. The introduction will end with a small report structure overview that will explain each section of the report.

## 1.1 Motivation and Context

This project covers the case 'Human Behavior at the Library', which collaborates with the Library of The University of Southern Denmark, Odense. Previous investigations made within the library revealed the migration of visitors within specifics of which entrances of the library do they tend to use to enter and leave. The aim is to gain additional information about visitors and their movement. This might help the library properly optimize how to rearrange the rooms, change the opening hours, make alternative routes available or edit existing ones.

An IoT solution is a perfect match for this problem. If the solution to gathering this data was purely a smartphone app, then there might not be many that would participate, definitely not if the app did not provide any immediate value to the library visitor, who the library wants to know the movement of. An IoT solution has the advantage of being less direct on the library users.

## 1.2 Objective

The objective of this library case, is to produce some system that can gather information valuable to the library, about the library visitors. Which could be knowledge about how visitors tend to move around inside the library. To do this, we are going to design and implement an Internet of things system. Our case also covers a system property, that we would like to focus on - Security.

## 1.3 Resources

The project has been given two IoT hardware kits. The hardware given is the Arrow SmartEverything IoT Development Boards. A short list of its specifications reads as.
**Communications:**

- SIGFOX.
- Bluetooth Low Energy (BLE).
- Near Field Communications (NFC).

**Sensors:**

- GPS as the GNSS.
- Accelerometer, gyro, and magnometer.
- Humidity and temperature.
- Barometer.
- Light sensor.

## 1.4 Report outline

Following is a short outline of the purposes of the reports different sections are, and what the reader can expect.

**Introduction**
Provides overview and an understanding of the immediate domain.

**Problem Description**
Evaluates and defines the problem in a way that makes it approachable and hopefully solvable.

**Solution Approach**
Asks what is the idea for a solution to the before stated problem and possible hurdles.

**Analysis**
Gives a more in-depth analysis on the what/how of the system, looking into use cases and requirements.

**Design**
Explains and visualizes some of the systems design, for example how the system is deployed.

**Implementation**
Describes the code developed for the IoT board as well as the android application that connects the boards data with the cloud storage.

**Results and Evaluation**
Presents data/results gathered using the system, and evaluates the system based on this.

**Discussion**
Debates the challenges that has been encountered designing and developing an IoT system.

**Conclusion**
Summarizing the most important aspects of all the work in a concluding answer to the described problem.

**Future Work**
Ideas for improvements and new features that the group would work on next, should the project continue.

# 2 Problem Description

The library of SDU has an interest in gaining knowledge about the movement patterns of the people visiting the library, while also valuing the privacy and confidence of the individuals. Interesting perspectives to be looked at includes:[1]

- Where do people tend to gather in the library, at what time of the day?
- Are there any patterns in terms of how visitors tend to move around inside the library space
- What can be said about the time spend in the library

From this it is very clear what the overall theme is. So the data that is going to be gathered, should be able to tell the library about its users usage of their facilities.

Beyond this, the group has also chosen to have security as an important parameter. The security focus of the project should also target: Physical security, Device control security, Connectivity security and Data security.

Based on this, the problem defined for this project is: How can a secure IoT system that gathers data about library visitors movements be developed?

# 3 Solution Approach

Two approaches were seriously considered by the group. Both will be talked about in this section, and why the first approach was picked.

The idea is to implement a program on the SmartEverything IoT device, that is able to track the movement of its carrier, by the use of GPS. The tracking should be able to happen without further requirements to the carrier, than that they hold the device on their body. The device is to be handed out at entrances, where it is set off by the person handing it out - either by the reset

button, or interaction with a smartphone and an application. This handover and participation might be further incentivised by giving cookies or other treats at the return of the device. The data has to be offloaded by some means, either during tracking in bits, or at the end of the persons library visit in one total package. With this data, heatmaps or paths of movement could be made and visualized for the library.

Another approach was also in the talks, before the previously discussed was picked. This approach would have the sensors placed at the entrances, registering the doors being used, potentially if an entrance or an exit was made. The data gathered from this approach would at its best have been how many are in the library at what times, and what exits, if enough sensors to cover all exits is available, are used and when. This data is not without value to the SDU Library, But besides the exits and entrance usage, there would be no data on the movement patterns and library traversal. This is the main reason we did not attempt to implement this approach. The second reason being an uncertainty towards being able to detect whether an exit or entrance was made, using the given IoT board. Specifically, we were not sure how precise the data from the proximity detector and light sensor would be, and if we could precisely determine exits and entrances separately from them.

# 4 Analysis

This section analyses the what/how of the system, looking into use cases and requirements.

We have been given two SmartEveryThing hardware boards, as mentioned in the Objective section. After reading documentation and user guide[2], we gained understanding of the sensors included included in the hardware device. We found three relevant hardware sensor options that could be used to solve the initial problems: GPS Module with Embedded Antenna and Proximity sensor. Accelerometer sensor could also be up to consideration.

Proximity and light sensing module can be used to gather information about where do people tend to gather in the library and at what time if placed in the study rooms. We can also get information about the time spend in the library. We could deduce which exits people use - where do these exits lead to. We could find out which times of the day are the peak hours for different sections of the building and which are used to the lesser extend.

With the use of GPS, or alternatively with accelerometer, we can get a more telling information movement of the people as we would like to. We can track patterns and see which of the corridors, study rooms and gathering checkpoints are used more frequently then others. We can get data about the time spend in the library.

In the solution approach, we presented two ideas. One, which we want to give more merit based on the actual results that could be given as an outcome.

Consultation with other student groups that work on similar problems revealed preferred use of proximity sensor in projects to see where and when people gather within the library and which entrances and exits are used more. As we would like to contribute with data that could still be beneficial library of SDU, we decided to look at the problem from other perspective. For this, we will be using GPS module.

## 4.1 Actors

There are only two significant factors in the system we would like to design. The main factor - the people visiting library and the secondary factor of people in charge of the distribution of the hardware and monitoring the misuse and problems. In view of this, we can write that we have to categories:

- Library visitor
- Library worker

We wrote Library worker, however this category can be also represented by student volunteers or the people most familiar with the system, our team.

## 4.2 Use Case

With the actors from the previous subsection, we created use cases to show what our prototype and system should offer.

- UC1: Collection of data from IoT device
- UC2: Offloading data to the android application
- UC3: Saving the data to the database
- UC4: Visualization

We demonstrate different ways user might interact and the basic flow of our system on Figure 4.1. This is not how the end product will look like, but an abstract draft, the first high-level view of the system.

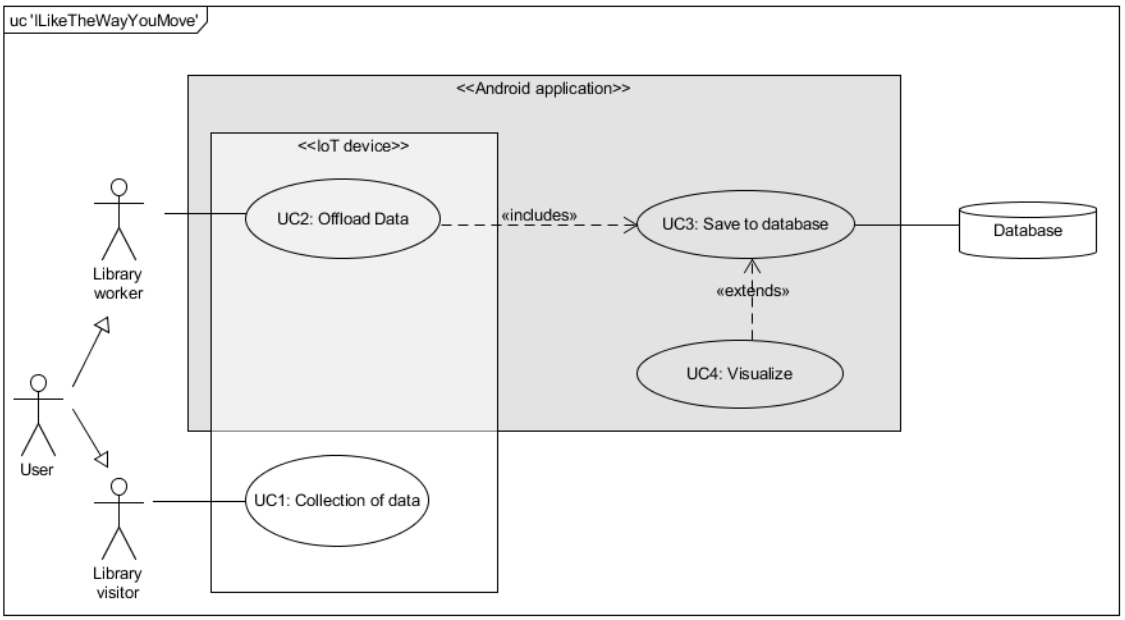


Figure 4.1: Project Use Case

## 4.3 Requirements

This subsections gives description of the set requirements. They were created in association with the use cases written in previous subsection, the task given by our supervisor and on the education and understanding of system development within our group.

### 4.3.1 Functional requirements

- The User should be able to start and end collection of movement data
- The Library worker should be able to register into, login and logout of the Mobile Application
- The Library worker should be able to offload data from IoT device to Mobile Application on Mobile device
- Ability to save data to database

- Visualization of the results

### 4.3.2 Nonfunctional requirements

- Use of at least one sensor to collect data
- Use of at least one radio to transmit data
- Focus on security aspect
- Energy efficiency
- Mobile Application is compatible with Android system
- UI of the Mobile application is user friendly

## 4.4 Data gathering

For data gathering of the location the possibilities provided by the hardware are using GPS data or accelerometer data. As the library is indoors the signal strength of the GPS is a big concern as it could have a big impact on the accuracy of the data. Accelerometer is also an option, as the starting location is known, it could be possible to estimate the route based on the accelerometer data.

## 4.5 Data offloading

To retrieve data from the devices, the arduino offers SIGFOX, Bluetooth and NFC. SIGFOX supports up to 140 uplink messages a day, each containing a maximum of a 12 bytes payload.[3] This makes it unsuitable for constantly sending location information, as it would quickly reach the 140 messages limit. For this using bluetooth would be more appropriate, but it would require the receiving device to be constantly within the range of the arduino device.

As real-time data is not important in this case, the data could just be kept on the arduino during the each tracking session, and then the data from the whole session could be offloaded at the end with the use of NFC. This could help the whole data collecting process to be user friendly, as the handler could just take the device, register it with NFC to signal the start a new session, register again with the NFC to signal end of session, and all the data would be offloaded and the arduino would be ready to start a new session.

This possibility with the NFC is also possible with Bluetooth, but not with SIGFOX, and there would be too much data.

## 4.6 Sensing framework

| Categories | ILikeTheWayYouMove |
|---|---|
| Information Type | Presence |
| Occupant Relation | Anonymous |
| Sensing Strategy | Augment persons |
| Spatial Granularity | Space |
| Temporal Granularity | Periodic |
| Spatial Coverage | Space |
| Temporal Coverage | Past |
| Sensor Modality | EM waves |
| Methods and Models | |

Table 1: Categorizations for ILikeTheWayYouMove system

Using the framework proposed by Kjaergaard et al[4] we have classified the different options available for the implementation of the system, to help make the best decisions based on the pros and cons. All the categorizations can be seen in table 1.

The **information type** needed is *Presence*, as the goal is to plot the locations of the occupants moving through the library. The **occupant relation** of the system should be *anonymous* in our case, as it is not the goal to analyze specific individuals in the library, but get a general idea of the overall movement patterns. This removes the need for collecting any personal data from the occupants, and so the privacy properties of the system would be easier to comply to. The **sensing strategy** to be used is the **augment persons** strategy, as the occupants are asked to carry an arduino device to track the location data. The **spatial granularity** is a *Space* as the system only concerns the library part of the building. The **temporal granularity** is *periodic*, as the arduino device periodically senses the location while the occupant carries it around. The **temporal coverage** of the system is *Past* as the location information is only stored locally while an occupant carries the arduino around, and the information is not available until the data has been offloaded. It would be possible to make it *Present*, if the data was constantly offloaded instead of stored locally, by bluetooth or SIGFOX, but as the goal of this system is not to track the current visitors of the library in real-time rather than to gather information about their movement patterns, that can latter be analyzed upon, it is not a relevant feature. The system making use of GPS makes the **Sensor modality** *EM waves*. This provides good mapping to spatial properties, but the accuracy can be highly influenced by the environment and the presence of humans, which in this situation are both relevant, as the tracking happens indoor in a building in a library where there is most likely at least a few other people around. Therefor to make the tracking more accurate another sensor modality could be introduced. *Force* could be used to make the tracking more accurate with the use of accelerometer.

Had we wanted to create a system that would tell us how the occupants of the library spend their time in the library during specific times, e.g. lunch break, study time, just meeting some people there, we could use *Conditional rules* model. This way, we could let the users select an option of the reason behind their stay in the library before starting tracking their movement. It would give us the opportunity to see differences in the length of their stay and practices for various behavior types.

# 5 Design

This section describes the design decisions that are to be taken into consideration during later implementation phase. We describe the architecture of the system, a tactic to be used in quality attribute scenario and we also describe an User Interface and application flow of the mobile application.

## 5.1 Deployment diagram

Deployment diagram on Figure 5.1 shows the necessary computation units that our system requires.

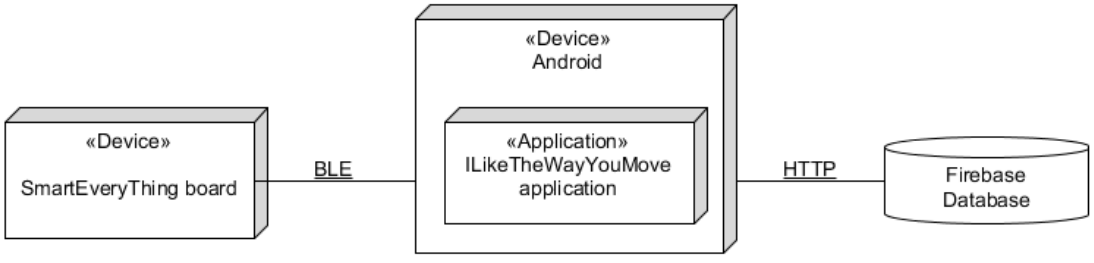


Figure 5.1: Deployment diagram

The minimal amount of units is two. A Mobile device with access to a database and an IoT device.

Two **SmartEveryThing** boards communicate through **BLE** connection with a Mobile Device.

A mobile device hosts a **Mobile Application** *ILikeTheWayYouMove*, that communicates with a **Firebase** Database via **HTTP** protocol.

## 5.2 Quality attribute scenario

In order to increase the quality of the system, and to fulfill our functional requirement for the system to be energy efficient, we suggest the following quality attribute scenario displayed on Table 2.

| Source of Stimulus | User |
|---|---|
| Stimulus | Accelerometer recognizes movement activity |
| Environment | Normal environment |
| Artifact | SmartEveryThing board |
| Response | Starts GPS tracking |
| Response Measure | Less battery drain |

Table 2: Quality attribute scenario

For this reason, we present the tactic **Sensor Replacement** proposed by Kjaergaard et al[5]. This tactic suggests using the least costly sensor to trigger another one. In our case, it would be Accelerometer sensor triggering GPS module. With it, when the user is sitting by the table for a long time, we do not waste our resources by picking up the same data repeatedly. Only once he moves to travel to another destination does the GPS start again.

## 5.3 Mobile application User Interface

The mobile application consists from two activities. The first one is the LoginActivity where the users use to login to the application and the second one is the MainActivity. In the last activity users are able to see the status of BLE and GPS. In addition, they can change the MAC address to connect to each device, refresh the connection and log out from the application. Below there are two images, the first is the LoginActivity and the other the MainActivity.

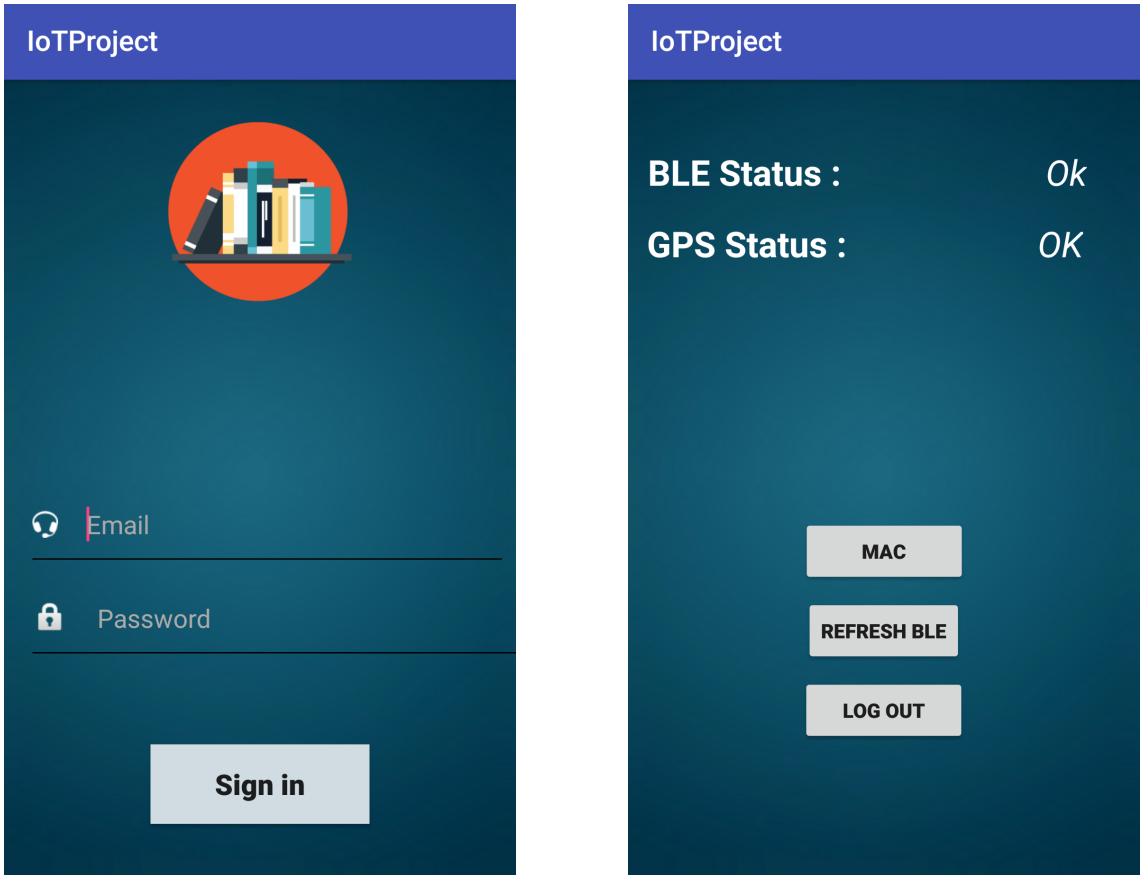


Figure 5.2: The Login and Main Screens.

## 5.4 The data gathered and its visualization

The data that is intended to be gathered is GPS positions and nothing else. The library does not want to collect any data that might be of too personal a nature, and so no other type of data gathering was to be implemented.

Once the data is gathered, it is offloaded into a mobile application through BLE. This is done as SIGFOX is too limiting to be the data offloading connection the system needs. From the application, the data is sent to Google's Firebase, where it is stored.

From here it is then the idea to process the data depending on the quality of data that is achieved, as well as to what purpose it is needed for. Most likely some filtering techniques would be necessary, and possibly also some simplification, for efficiency. An optimal end result would use the data of many gathering sessions, creating either a heat map, or paths with elements of heat in them as well, as shown in Figure 5.3.


Figure 5.3: Digital recreation of how the group would like the end result of the data visualization to be.

What Figure 5.3 shows, is a part of the library, often referred to as the technical library, that has both many sections and entrances. Pathways used, as well as the pathways in between the book sections are highlighted. The color on the highlighting then depends on the heat of that pathway, i.e. how busy that pathway is. If this visualization could be delivered to the library on a dynamic map, where an hour at a time is visualized, and the library could change between what time of day and what day the wished to see, then that would deliver the most value to the library, that we see possible with the GPS data.

The time aspect has not been designed into the developed IoT solution, but might just be noted down with the data offloading to the app, as it is then uploaded to the cloud storage.

From this, the library would now quite precisely how it's facilities is used, as they themselves know how it is arranged, and at what days and time they are used. They would know this without infringing on any personal information of their library users, but of course with the hurdle of active participation, in the form of library users carrying around the IoT devices.

# 6 Implementation

In this section, the implemented software will be deliberated. The most important/interesting parts of the developed Arduino- and Android code will be presented and shortly discussed.

## 6.1 Arduino Code

In this chapter we write about the Arduino code that we have developed in order to make the Smarteveryhting device functional. The code consists from three functions, the setup, the loop and the double2string.

In the first one we initialize and start the serialUSB, GPS, BLE and accelerometer in order to use them later. In the loop function there is all the main functionality of the application. Firstly, we check if the GPS sensor is ready for use, then if it is the right time and at the end if the accelerometer's values have changed. If everything is true then we use smeGPS to gather GPS data, then format them on the proper type using the double2string function and send them via BLE to the smartphone.

In order to implement all the above functionalities, we have used the Smarteverything's libraries.[6] Below we present a part of the Arduino code.

```
if (smeGps.ready()) {
       if ((loop_cnt % 2000) == 0) {
         if((abs((x-smeAccelerometer.readX()))>5)||(abs((y-smeAccelerometer.readY()))
          ↪  >5)||(abs((z-smeAccelerometer.readZ()))>5)){
             x = smeAccelerometer.readX();
             y = smeAccelerometer.readY();
             z = smeAccelerometer.readZ();
             altitude   = smeGps.getAltitude();
             latitude   = smeGps.getLatitude();
             longitude  = smeGps.getLongitude();
             String latitudeString = double2string(latitude,10000000);
             String longitudeString = double2string(longitude,10000000);
             String altitudeString = double2string(altitude,10000000);
             char latitudechar[latitudeString.length()+1];
             char longitudechar[longitudeString.length()+1];
             char altitudechar[altitudeString.length()+1];
             (latitudeString).toCharArray(latitudechar,sizeof(latitudechar));
             (longitudeString).toCharArray(longitudechar,longitudeString.length());
             (altitudeString).toCharArray(altitudechar,altitudeString.length());
             String allthedatastring = latitudeString + "=" +longitudeString;
             char allthedatachar[allthedatastring.length()];
             allthedatastring.toCharArray(allthedatachar,allthedatastring.length());
             smeBle.write(allthedatachar,sizeof(allthedatachar));
             Serial.println(allthedatachar);
       }else{
           x = smeAccelerometer.readX();
           y = smeAccelerometer.readY();
           z = smeAccelerometer.readZ();
       }
      }
   }
```

Code Snippet 1: Loop function in Arduino code.

## 6.2 Android Code

In order to offload the data from the arduino, and get it uploaded to a Firebase database, we have developed an android app.

As seen in code snippet 2 when the App has succesfully connected to the Arduino, it will call **read()** (line 11). This method (see code snippet 3) sets up an **BleNotifyCallback**, so that when the Arduino writes out an message using BLE, the app will recieve it and handle it as seen in snippet 4.

Bellow is the string recieved from the Arduino that is converted into lat and long coordinates, and then using the method **addToDatabase** is pushed the database, as seen in snippet 5.

```
private void Connect() {
        ...
        BleManager.getInstance().connect(MAC, new BleGattCallback() {
                @Override
                public void onStartConnect() {
                        bleTextView.setText(R.string.wait);
                }
                @Override
                public void onConnectSuccess(BleDevice bleDevice, BluetoothGatt gatt,
                 ↪  int status) {
                        myDevice = bleDevice;
                        bleTextView.setText(R.string.ok);
                        read();
                }
                ...
        }
    ...
}
```

Code Snippet 2: Connection to the Arduino using BLE

```
BleManager.getInstance().notify(myDevice, servUUID, readUUID,
new BleNotifyCallback() {
        ...
        @Override
        public void onCharacteristicChanged(final byte[] data) {
        ...
    }
        ...
}
```

Code Snippet 3: Setup of BleNotifyCallback()

```
public void onCharacteristicChanged(final byte[] data) {
        ...
        prevData = str;
        String lat, lon;
        int eq = str.indexOf('=');
        lat = str.substring(0, eq);
        lon = str.substring(eq + 1);//xoris to =
        DecimalFormat df = new DecimalFormat("#.######");
        lat = Double.valueOf(df.format(Double.parseDouble(lat))) + ""; // rounding
        ↪  the numbers (restoring to the original number)
        lon = Double.valueOf(df.format(Double.parseDouble(lon))) + ""; // rounding
        ↪  the numbers (restoring to the original number)
        gpsTextView.setText(R.string.ok);
        FireBaseClass.addToDataBase(lat, lon);
        ...
}
```

Code Snippet 4: Handling of message from Arduino

```
public static boolean addToDataBase(String lat, String lon) {
        ...
        currentFirebaseUser = FirebaseAuth.getInstance().getCurrentUser();
        database = FirebaseDatabase.getInstance();
        Map<String, String> userData = new HashMap<>();
        userData.put("Lat", lat + "");
        userData.put("Lon", lon);
        DatabaseReference myRef =
        ↪  database.getReference().child(currentFirebaseUser.getUid());
        myRef.push().setValue(userData);
        return true;
        ...
}
```

Code Snippet 5: Add to Firebase database

## 6.3 Implementation Troubles

Before the discussion, we would here like to discuss some problems that we had with implementing the presented IoT system. As the troubles were many times of a very technical nature, we find it more fitting as an end to this section, than the discussion section.

Firstly, we had to choose the most efficient way to establish a connection between the Smarteverything system and the android phone. To achieve this, we use the FastBle open source library. We have chosen this one because it is the most reliable open source library for this purpose and makes the developing part of the android application more efficient.

The next thing was to find the right UUIDs that Smarteverything device uses. We need the general one to establish the connection and the specific one to send the data through BLE. To find the UUIDs we used the mobile application B-BLE that we find in the documentation of the BLE sensor of Smarteverything.

One of the problems that we were against was the conversion of a double type number to an array of char. Normally when coding for Arduino it is simple to do this kind for conversion but for Smarteverything the normal way is not working. Thus, we used the method “double2string” that we found in the Smarteverything’s repositories on Github. The problem with this method is that sometimes the numbers could change a little for instance from 51.123456 to 51.1234559. In the end, we found out that the application rounds the numbers and we solved it by setting the precision to 6 to retrieve the original number.

One of the problems with the IOT devices is the amount of energy that they consume so we had also to consider about it. In order to make the device last longer we use at first place the accelerometer sensor to check if the user moves or not. If they move then we retrieve the GPS data and send them over BLE to the smartphone else the device waits till any movement. This was done as written in the Design section.

These were the major problems that we were against with. We have found an optimal solution for all of them in order to develop a functional and efficient system.

# 7 Results and Evaluation

In order to estimate application’s effectiveness and accuracy, the board and the developed software has been put to use in different environments. In this chapter we present and evaluate the results that we have gathered.

Of the two boards that has been available, it seems the GPS functionality has been impaired. The board was taken to be tested inside the library, and although we at that point in testing did expect problems with indoor tracking, the results was a total lack of data. The board was then taken outside, were it had been observed that the other board worked very well, but still there was a total lack of GPS data coming in. The other board, that had been observed to work was in Copenhagen, and would not be back in Odense before after the due date of this reports hand-in. Instead, the working board was put to test in Copenhagen. In a ”highly” indoor setting, meaning far from windows and in a big building, we were unable to gain meaningful data. In addition to this, the device was tested in another smaller indoor environment closer to a window. Here data was received, but the data were of a less accurate nature. In the Figure 7.1 the blue spot presents the right location meanwhile the red spot presents the location that the device located. It is obvious that for a project like this even the few meters might be important so this margin of error it is not acceptable.

The device was lastly taken on an outside walk in Copenhagen, where data was gathered. This data has been visualized in Google Maps and is presented in Figure 7.2.

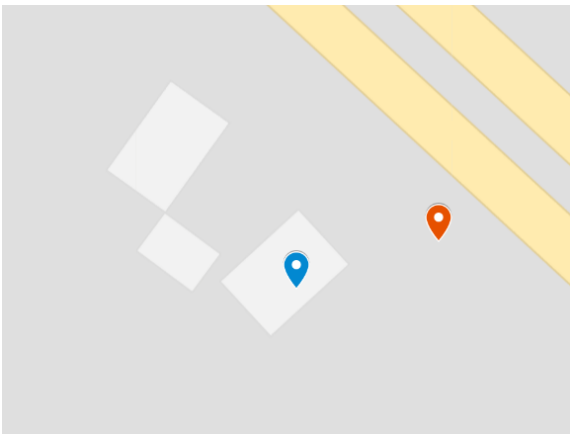

Figure 7.1: Indoor tracking

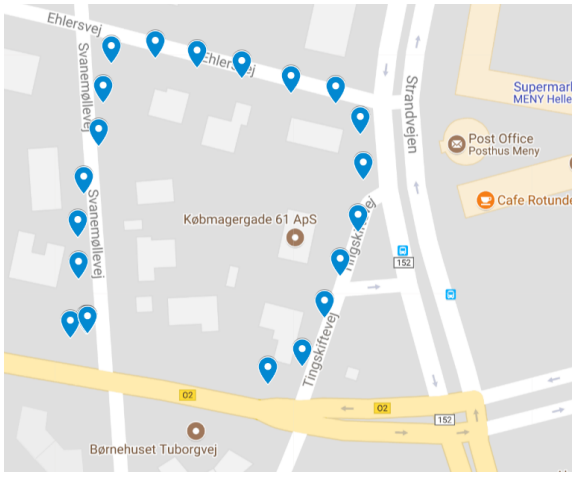


Figure 7.2: Outdoor tracking

This matches very well with the walk taken, to the degree of precision that would have been wanted to tracking inside the SDU Library.

# 8 Discussion

Our findings in previous section Results and Evaluation **refute the initial solution approach theory** that with the use of GPS, the device we used is able to track movement of its carrier. This would be only true, if the movement was happening outside the building, not inside. The quality of data gathered inside is insufficient for the purpose of further processing. The signal is very weak, buried in white nose.

It is also important to point out that even if we had successfully gathered data we desired from GPS, there would still be several additional issues to be investigated.

## 8.1 The choice of used methods and sensing strategy

One of them is the choice to use **augment persons sensing strategy**, that we have elected in the Sensing framework subsection of Analysis. As stated in [4], this strategy augments people with wearables or mobile sensor systems to gather information. For many users, wearing on-body sensors or sensors carried on their hand the whole time is a major *inconvenience*. A solution to this might be to build such sensors directly into devices that the users already use. A mobile device is an example. With this solution would however come a disturbance in the user concealment and the potential *security concern* as the user would need to install an application on their personal phone.

The main reason as to why we have chosen this strategy was the told advantage of collecting detailed data about the individual user physical movement. As the results in our case are not adequate, this advantage in our solution approach has become redundant. Alternative solution could be to use different sensors to gather data or use other sensing strategy.

Alternative sensors for data collection could be to use BLE Beacons, integrate linear acceleration to get position using gyroscope and accelerometer or use of sensor-fusion. Another suggestion would be to use a device with support to WAAS[7]. With its addition to GPS, it would give a correction with the accuracy of 3 meters 95 percent of time. WAAS is however supported only in North America. In Europe, EGNOS could be considered.

A drawback to using our strategy is also the collection of data on a large scale. It would either require a lot of time collecting with limited recourses of arduino devices or the added potential cost for equipment. The extent of data we would need to have to be beneficial to the library would

need to have a lot of devices for such collection at disposal. And with that would come a *security issue* of how to secure the devices against theft and damage on larger scale. Therefore, a different sensing strategy is advised.

A way to reduce both the amount of devices needed and time required would be to change the sensing strategy from augment persons to **augment the environment**. This way only an amount of devices to cover the whole space would be needed. This however could still be a great amount depending on the total size of the area. If we would limit the area to only specific study rooms, entrances or any special space, it could prove to be a good alteration. This strategy would help collect data from every visitor using the library and not only selected or volunteered individuals carrying the arduino devices.

## 8.2 Security focus

In Problem description section, we wrote that this group has chosen a to have **security** an important aspect to look at. And we would like to discuss here how we approached this problem during the whole design and development of the solution. There are four security features we focus on. Physical security, Device control security, Connectivity security and a Data security.

- **Data security**

  While collecting data, we gather only information relevant to the position of the user. We do not amass any personal information that would give away the identity of the visitor. It is completely anonymous.

- **Physical security**

  There is a surveillance team in task of monitoring the physical movement of the device. It consists of people in charge of data collection and our suggestion is to place them at the entrances and exits in the areas of interest. A security feature against theft and lost could be using GPS to track the device even outside of collection of relevant data.

  The arduino device that we were given included a hard cover made of plastic to lessen damage of the device in case of crash. We would like to reduce the harm more by placing a material inside the cover, wrapping the device so that there is less space for it to move and to add cushion.

- **Device control security**

  The people in surveillance team are also the only ones able to sign in and access the mobile application by the way of authentication activity inside the application. Another solution to device control security question would be to encrypt data transferred from and to devices to prevent unauthorized use or dissemination. This could be fulfilled by implementing SIGFOX. It has implemented a limited amount of data and types that can be transferred.

- **Connectivity security**

  The communication between arduino and mobile application is based on BLE. The mobile app only connects with one specific arduino board at a time, based on its MAC specification.

We have also already addressed some additional security concerns in suggested alternatives to the solution in previous subsection.

# 9 Conclusion

From the three proposed perspectives in the problem description, we have chosen the second one: *"Are there any patterns in terms of how visitors tend to move around inside the library space?"*. We aimed our attention to see how the data can be gathered and to which conclusions we can come to with the results.

The security focus parameter that the group have chosen has been implemented and in the discussion, we considered additional measures that can be taken to secure the system.

For the project requirements to use at least one sensor to collect data, we have chosen a GPS module together with Accelerometer sensor to lessen power drain. The demand to use at least one radio to transmit data have been fulfilled by using BLE network. As discussed in previous section, a SIGFOX implementation can be considered have a more secure data connection and transmission.

We have implemented a prototype, that can collect GPS data, send it to the mobile application and then to Firebase database. At present state, we do not have a visualization feature, or a data processing option to help us better categorize the data and use it accordingly.

Although the now documented approach is a solution, if a boards GPS receiver is strong enough, we do not recommend it as a practical solution to the library's problem.

# 10 Future Work

In this part we discuss about improvements that could be made on the project to improve efficiency, accuracy and learning outcome.

With the present structure of the project users have no benefit from it so it could be difficult to convince them to use it. Thus, we should make the mobile application which would be somehow also useful for the user. We could achieve this if the application provides general information about the library like temperature, humidity, pressure and an approximation of the present number of visitors.

The next thing that could increase project's usability is to change MAC address input's method. With the present way users must input the MAC address themselves, but a more efficient way could be using NFC to send the address from the Smarteverything device to the smartphone. From the other point of view, we could also make improvements on the backend of the project. We believe that the most important improvements are to increase the accuracy and to find an optimal way of storing the data in the database.

A way to increase accuracy is by putting BLE Beacons in the library and calculate user's location by them.

Moreover, the way of storing the data is crucial because based on that we might have better results. For instance, if we also store the timestamp of each input we could extract more useful information about where users spend more time in the library.

In conclusion, the present edition of the project is completely functional but there are many things that could be improved in the future.

# References


[1] Mikkel Baun Kjærgaard. Project ideas within internet of things for the uci course at sdu. https://e-learn.sdu.dk/bbcswebdav/courses/SM2-UCI-U1-1-F18/Materials/Lecture%205/Project%20Ideas%20within%20Internet%20of%20Things%20for%20the%20UCI%20course%20at%20SDU.pdf. Accessed: 2018-04-20.

[2] Amel Technology. Smarteverything fox, user guide. http://docs-europe.electrocomponents.com/webdocs/144b/0900766b8144b09d.pdf. Accessed: 2018-05-21.

[3] Wikipedia. Sigfox. https://en.wikipedia.org/wiki/Sigfox. Accessed: 2018-05-25.

[4] Fisayo Caleb Sangogboye Mikkel Baun Kjærgaard. Categorization framework and survey of occupancy sensing systems. https://e-learn.sdu.dk/bbcswebdav/courses/SM2-UCI-U1-1-F18/Materials/Lecture%205/Kjaergaard%20%282017%29%20-%

20Categorization%20and%20Survey%20of%20Occupancy%20Sensing%20Systems.pdf. Accessed: 2018-05-20.

[5] Marco Kjærgaard, Mikkel Baun; Kuhrmann. On architectural qualities and tactics for mobile sensing. http://findresearcher.sdu.dk/portal/files/112530464/mobilesensingtactics.pdf. Accessed: 2018-05-21.

[6] SmartEverything. sme-se868-a-library, sme-vl6180x-library, sme-cc2541-library, sme-lsm9ds1-library, sme-arduino-core. https://github.com/ameltech.

[7] Garmin. What is waas? https://www8.garmin.com/aboutGPS/waas.html. Accessed: 2018-05-22.

[8] Jasonchenlijian. Fastble. https://github.com/Jasonchenlijian/FastBle.