UiT The Arctic University of Norway

Department of Industrial Engineering

Building Information Modelling: Indoor localization

Jeewan Khadka

Master's thesis in Industrial Engineering   INE-3900 15 May 2021

Acknowledgements

I would like to express my sincere gratitude to my supervisor Gabor Sziebig, Senior Research Scientist and Associate Professor at UiT Narvik, for his constructive guidance and practical recommendations throughout my thesis. I am particularly grateful to Nils Andreassen, Senior Engineer in the department for assisting me with the installation part. I am equally grateful to other helpful staffs in the faculty who provided me the necessary suggestions and supports in need.

I would also like to extend my thanks to Mr. Sujan Maharjan, PhD candidate in the department. for his inspiration and suggestions on my work. Similarly, I express my gratitude to Mr. Kwan Thanapong for volunteering in localization test. Also, thanks to several open-source platforms and their participants for entertaining my queries and helping me to get through several technical issues.

Furthermore, my deepest thanks go to my family, and my friends for their love, understanding and continuous support to during this period. My thanks are incomplete without addressing those who directly or indirectly supported me I am very grateful to them.

-Jeewan Khadka

Table of Contents

List of Tables

List of Figures

List of Abbreviation

Abbreviation	**Explanation**
2D	2 Dimensional
3D	3 Dimensional
ACK / NACK	Acknowledgement / Negative Acknowledgement
ADO	Active X Data Object
AoA	Angle of Arrival
AOPT	Analyzing and Optimizing
API	Application Programming Interface
BIM	Building Information Modelling
BLE	Bluetooth Low Energy
CAD	Computer Aided Design
CSV	Comma Seperated Value
dB	Decibel
dBm	Decibel milliwatt
EB	Echobeacon
EM	Electromagnetic
GPS	Global Positioning System
H&S	Health & Safety
HTTP	Hyper Transfer Protocol
I2C	Inter-Integrated Chip
IAAS	Infrastructure as a Service
Id	Identification
IEEE	Institute of Electrical and Electronics Engineers
IFC	Industry Foundation class
IoT	Internet of Things
IP	Indoor Positioning
IR	Infrared Radiation
ISM	Industrial Scientific Medical
JSON	JavaScript Object Notation
LAN	Local Area Network
Mbps	Mega byte per second
MQTT	Message Queuing Telementry Transport

MS	Microsoft
ms	miliseconds
NET	Network Enable Technology
NW	Navisworks
NWC	Navisworks Cache
NWD	Naviswork document
ODBC	Open Database Connectivity
O-DF	Open Data Format
O-MI	Open Messaging Interface
PAAS	Platform as a Service
PC	Personal Computer
RF	Radio Frequency
RFID	Radio Frequency Identification
RSS	Received signal strength
RSSI	Received signal strength Indicator
SAAS	Software as a Service
SDK	Software Development Kit
SOA	Service Oriented Architecture
SQL	Structured Query Language
SVG	Scalable Vector Graphics
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TDoA	Time Difference of Arrival
ToA	Time of Arrival
TOG	The Open Group
UHF	Ultra High Frequency
URL	Unique Resource Locator
UWB	Ultra Wide Band
WEBGL	Web Graphics Library
WIFI	Wireless fidelity
WLAN	Wireless LAN
XML	Extensive Markup Language

Foreword

One of the main focus of Architecture Engineering and Construction (AEC) industry today is to minimize the cost for construction and maintenance of buildings without compromising quality. Tracking of resources and work progress in the construction phase can help to efficiently plan and manage inventory saving the costs. Similarly, monitoring indoor environment and people movements provide overview on occupancy and utilization of facilities. Facility managers can track such information for efficient utilization & optimization of energy consumptions. In construction environment awareness on physical environment and information on people positioning can be used to minimize worker's risk of exposure to dangerous situations. Similarly, locating assets & people in industrial area or larger spaces like warehouse, hospitals, universities, and public buildings can be effectively carried out if the positioning information are available, this could save time of search which is even more crucial in emergency conditions. All of these physical data on locations, status & environment can be measured using IoT devices. Therefore, use of such devices to gather data and provide information in an intuitive way is motivating factor for this thesis. Architecture Engineering and Construction industries has been utilizing BIM methodology to model building works and share related information using 3D based design tools. The tools can be used to simulate over all construction process and include virtual models of the physical buildings with all related information required for its construction. These tools together with IoT devices have the potential to be used as a source to generate, process, and consume real time information.

This thesis presents similar system where the BIM software is used together with IoT devices to visualize data generated in real-time. Two different IoT devices are modeled as case study which measure environmental and localization data. These devices were installed inside a Test room of 22 m^2 . The collected data were, filtered & transferred to database server which were then retrieved and visualized by BIM software in real time. The report presents tools and technologies that are implemented to develop such system and provides details on the basic blocks required for such integrations. The combined platform provides information about the things as it happens in real-time. This makes such systems capable for digitalization of physical process and have various application domains. In the report it is applied as monitoring platform for temperature and illumination data and can be used for facility management applications. Similarly, indoor localization is monitored making it applicable for localization and safety management purpose. The performance of the system is also discussed based on test, observations, and calculations.

I. Problem Statement

When it comes to safety and facility management, it is crucial to have a monitoring platform that provides information on field devices, environmental data, resources, and personnel location. Development of a platform to provide both information on environmental data and indoor localization can help in maintaining safe and secure indoor conditions.

Traditional cctv camera may not provide sufficient information related to object tracking or resource localization. Also, in an indoor or underground construction environment the deployment of workers is often planned but during work execution information on workers movement and site conditions can change making it hard to predict. Poor site environment can have negative effect on workers' health. While lack of information on real time positions of people in an indoor area could make it harder for safety managers to develop disaster management and response plans during emergency situations. Thus, increasing chances for casualties. Furthermore, quick availability of information on location of resources and material in larger warehouse and industrial spaces can be beneficial for effective localization saving the time of search.

II. Project Description

The project is to present a model of an indoor area using BIM software and utilize IoT devices to collect localization and environment data. The data is to be visualized through an integrated platform of BIM and IoT.

Physical data can be transferred to virtual environment using IoT based communication protocol. A simple IoT framework as described in section 1.2 can be considered to understand the underlying principles for seamless data transfer to other platforms. It consist of basic four layers of blocks responsible for data gathering, sorting, transferring, processing, computation, management and publishing [1]. BIM software API, a software intermediary allowing abstraction of functionality between the system can help in enabling the integration and visualization of data to BIM platform.

There is need for development of an integrated platform that can support safe and secure indoor environment. Integrating BIM and IoT would help AEC industry with required technology for monitoring of environment and location tracking of resources and manpower. The awareness on surrounding and people movement can also support energy saving through facility management.

III. Objectives

The main objective of this project is to develop an integrated system using BIM and IoT tools that can be implemented for indoor positioning and monitoring of environmental data. The project would test the conceptual frameworks to develop an integrated system and study how the system works together. A case study is to be carried out using a test room available in UiT Narvik, campus premises. Also, findings are to be documented to confirm usability of the system.

IV. Report Structure

The report is divided in to four sections, where section 1, provides introduction on BIM & IoT integrated system, related works, and details on indoor positioning system. Section 2 describes methods for implementation of BIM & IoT tools to develop an integrated system, tools and technology used. It also presents details on how the case study is conducted and parameters considered for measuring performance of the system. In Section 3, results of the case study are presented, analysed, and discussed. Similarly, Section 4 presents conclusions of the report.

1 Related Research

Building Information Modelling (BIM) is a methodology used for modelling of design, construction and management related activities of building or infrastructure in Architecture, Engineering, and construction (AEC) sectors. It is based on 3D modelling techniques and uses 3D CAD softwares like Revit, ArchiCAD, Navisworks etc. BIM softwares are capable of incorporating all the building related information in 3D models that are required during project initiation to execution. Information sharing and coordination among several partners can be ensured using 3D models generated with BIM softwares. It is a helpful tool for clash detection and progress monitoring of construction related works. Using BIM together with sensors and IoT devices can collect several data associated with activities of the project to provide valuable insights on progress monitoring. Similarly, in built phases of the project, IoT devices can be used to collect environmental data that would help facility managers to monitor and develop strategy for comfortable indoor environment and energy optimizations. Information on localization of an asset or people in an indoor environment can be beneficial for large warehouse management or to develop strategy and plans for safe evacuation during the time of emergency. This report focuses on how use of BIM softwares together with IoT devices can be used as visualization platform for environmental data and indoor localizations. It also aims to highlight the requirements to develop such integrated system.

BIM models are shared to establish coordinated flow of information among several stakeholders in AEC industry. Standard file formats like IFC (Industry foundation class) can be generated using BIM softwares for information exchange which supports interoperability between different BIM softwares. Such standardization of file formats enables inter-exchange of information model with minimum loss of data. Real time design related information can be shared using IFC files however, for progress monitoring and information on status of facilities and resources in real time, needs devices that are capable to share actual site data to BIM model. The devices that are capable of enabling communication between things that are existent in real or virtual life can be utilized to generate and consume information. Such approach of focusing on state of “things” and on informing “things” about other things can be achieved by use of Internet of Things (IoT) devices [2]. Implementation of such devices for transferring of data and to BIM softwares can be advantageous for various applications to improve construction and operational efficiencies.

GPS (Global positioning System) technology is the most used localization and object tracking system presently. It is a satellite-based radio navigation system and are capable to functions under all weather conditions. However, for an indoor environment GPS signals are difficult to detect giving less accurate readings [3] . Some cellular network-based technologies are also used for localization and tracking of object. But changing transmission channels (EM-waves / Optical links) between receiving and sending ends results negative effect on wave propagation and alteration in signal strengths. So, these technologies are not reliable for all conditions [4]. This report presents a case study on using IoT devices for indoor localization where BIM tools are used as visualization platform. It provides details on Integrated system that uses BLE (Bluetooth Low energy) devices as IoT devices for collection of indoor localization data. Similarly, temperature and light sensors are used for collection of environmental data and integrated to BIM software for data visualization.

1.1 Building Information Modelling

BIM is used to generate virtual model of a real construction process including 3D models and enhanced information on various aspects that are useful for design collaboration, management, work execution and operation of projects [5]. Based upon the degree of information exchange it is categorized in to 4 different levels, level 0 to level 3 [6]. Level 0 includes 2D models of the construction objects and related information are shared in traditional means of papers or pdf versions. Level 1 BIM comprises 3D CAD & 2D drafting works, information is often managed by contractors and shared electronically. Similarly, in level 2 BIM all involved stakeholders collaborates in 3D model, common file format like industry foundation class (IFC) are used to share design information. The collaboration works however do not use single shared models in this level. In level 3 BIM, full collaboration between all parties can be done by using single shared project model. All stakeholders have access to common shared model which are made available in centralized repository (like cloud storage database) and can be used for modifications. Such method is also known as open BIM. At present most of the companies in AEC industries are found to be in level 0 to 2. The softwares used for BIM are capable of exporting models in various file formats among which IFC formats are standardized such that import, and export of files are supported by all the softwares used for realization of BIM in AEC sectors. Beside file transfer, these models can also be used to acquire information through shared resources such as model server database, by using APIs (Application Programming Interface) and web interfaces of the BIM server or database [2]. In such system IFC models acts as a single source for shared information among the participants. API is a software intermediary that allows abstraction of functionality between the system. It can be used to develop an app or platform to incorporate services provided by several softwares or webservers into a single application. So, by utilizing API of BIM softwares, cloud & webservers along with IoT a data visualization interface can be developed. Such integrated frameworks can offer the potential to function as "digital twins" of concerned built environment.

In AEC sector, Integration of BIM with cloud-based frameworks enables the usage and management of shared information between various stakeholders and software applications. Such cloud-based integration of BIM projects second generation of building information modelling BIM 2.0 [2, 7]. Where the First generation of BIM (BIM 1.0) are limited to shared information model or schema standard. while BIM 2.0 extends shared information into construction and project management methodology. This transformation of BIM requires to address four new dimensions i.e. enabling a) integrated environment of b)distributed information which is always c) up to date and open for d) derivation of new information [8].

1.2 Internet of Things

IoT for smart environment can be defined as interconnection of sensing and actuating devices providing the ability to share information across platforms through unified framework, developing a common operating picture for enabling innovative applications, that can be achieved by seamless ubiquitous sensing, data analytics and information representation with cloud computing as unifying frameworks [9]. To exploit power of cloud computing such IoT specific frameworks should be able to read data directly from the sensors or access from database, easily express data analysis logic as functions/operators for data processing and should have capability to present outcome to output streams that are connected to visual platforms. It is one of the fundamental technologies for digitalization & fourth industrial revolution as these devices are capable to generate and share information that are both crucial for functionality, remote access & decision making [10, 11].

IoT can be classified in to four categories based upon the usage such as Personal & home usage (individual), Enterprise (industries), utilities (service sectors) and mobile (smart transportations) [9]. One of the common applications of IoT is to track building utilities for environment monitoring. It has also been integral part of industrial automation, where actuators and sensors are used as a medium for communication, information processing or to trigger necessary set of instructions. Similarly, Implementation of IoT in Smart grid & smart meter, helps to keep track on energy consumptions of individual consumers and provide related information on load patterns. These information are used for efficient energy and resource management by the service providers. Such implementation of IoT helps in service optimization and are often based on cellular Wi-Fi or satellite communication networks .Thus, IoT can be categorized in to following four domains based upon their utilizations [12]:

a. Monitoring: It refers to tracking of devices conditions, environment state, notification, and alerts.
b. Control: It relates to control function of the devices and are used to actuate the system to obtain the desired output.
c. Optimization: Device performance, diagnostic and repair of devices is associated to optimization domain.
d. Autonomy: Devices used for autonomous operation of the system falls under this domain.

IoT devices have capability of communicating with each other where IoT middleware and cloud services plays vital role. Middleware allows users and applications to access data from interconnected objects and things, hiding internal communications and low level acquisitions aspects [13]. The general IoT reference architecture used for several IoT platforms consist of following elements[14] and is shown in Figure 1.

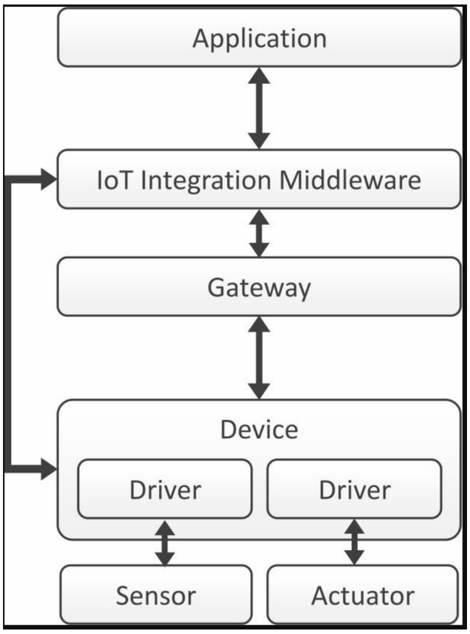


Figure 1 IoT reference architecture[14]

a. Sensors: These are components that measures environmental or physical parameters (data) and translate them to electrical signals. Usually, they are connected to microcontroller or single board devices (e.g., Arduino, Raspberry Pi etc.), which acts as an interface for data collection and transfer.

b. Actuators: These components receive commands or signal from the microcontroller devices to perform certain task so that desired condition can be achieved by manipulating physical parameters.

c. Device: It is a component connected to sensors and actuators that process data received from the sensors and controls the actuators. The sensors and actuators are either integrated or connected via wire or wirelessly. It requires softwares in the form of drivers to process data, transfer them or to provide signal to the actuators. These can be microcontroller-based devices or computers.

d. Gateway: A gateway serves as an interface that provides required facility to establish interconnection between communication protocol and technologies. Gateways helps in establishing the connection of the IoT devices with further system (database/ cloud services). It is also possible to filter, and process received data in the gateways before sending it to the further system.

e. IoT Integration Middleware: Middleware can receive data from the connected device directly or via gateway. It helps in evaluation of data based upon condition action rules such that it can be transferred to connected application and initiate control action required by devices through actuators or operator. Direct connection of the device with the middleware is possible if it supports appropriate communication technology, protocol, and compatible formats (Wi-Fi, HTTP or MQTT, JSON or XML). It serves as an integration layers for different sensors, actuators, devices, and applications. It can also provide functionalities like time series database, graphical dashboards, aggregation, and utilization of received data, user, and device management. It can be either software-based or cloud-based services.

f. Application: These are application software that are used to get overview of physical environment. It acts as a monitoring interface through which user can get access to the sensor data and control the environment using actuators. These applications are built upon various Middleware to have such access.

Increase use of IoT in various field projects the requirement of more storage and processing capabilities. Data generated from IoT devices need to be transferred into information and knowledge that can be applied to perform physical variations giving both economic and environmental benefits. Being low cost and light weight, it is not possible for IoT devices to locally store and process huge data. Therefore, there is need of the technology that are capable of more data processing and computation. Integration of IoT with cloud computing technology makes it capable for storing and processing of numerous data that can also be accessed remotely [15].

1.3 Cloud Computing

Cloud computing can be defined as a set of network enabled services, providing scalable quality of service guaranteed, normally personalized, inexpensive computing infrastructure on demand, which could be accessed in a simple and pervasive way [16]. The cloud computing services are delivered through internet and include tools and applications like data storage, databases, networking, and software interfaces. Some of the application characteristics of cloud computing can be mentioned as secure and dependable data storage center, ability of data sharing among various equipment, utmost possibility for internet use, easy to use and not requiring high quality equipment from user [17]. The cloud computing services are available for both public and private usage. These services have their own associated charges. Usually, the public services can be purchased based upon user requirements, while private services are hosted for specific clients. These services are delivered in 3 modes [18, 19],

a. Software as a service (SAAS): it allows user to connect to cloud-based application on demand through web interfaces or applications. These applications are usually run by the SAAS vendor over the data center and deliver functionality via internet to the user.

b. Platform as a service (PAAS): Unlike SAAS, PAAS allows the costumer to develop new applications and provides environment with computing resources for hosting the developed application. Here costumer have control over the deployed application and operating system.

c. Infrastructure as A Service (IAAS): Here the costumers are provided with virtual machines and other abstracted hardware and operating system. The costumers have control over storage, operating system, and application.

Some of the leading vendors providing these services are Microsoft Azure, Google cloud, Amazon web services, IBM Cloud, Alibaba cloud etc.

1.4 Literature review

The literature review is based upon the articles available in several data databases like Google Scholar, Science direct, Scopus and Oria. These articles are searched using relevant keywords like IoT & BIM integration, IoT localization, Cloud computing etc. Literature reviews is focused to several articles and works carried out using IoT and BIM integrated platforms and how these frameworks are developed.

1.4.1 BIM and IoT Integration

In article [20], authors presents a conceptual framework of generic BIM centered IoT. It consists of three basic elements to develop an integrated system. These elements are described below, and the framework is shown in Figure 2.

a. Network Physical Layer : This layer consists of physical set up for data accumulation. It includes sensors, objects, or surroundings under consideration. Data gathered are transferred to BIM-centered Middleware layer for further processing. The transmission of data is supported by the gateway or microcontroller devices based upon the communication protocols.

b. BIM-centered Middleware Layer: This layer receives the data from the Network physical layer to include it in BIM models. This layer can provide storage, data fusion and computation functionalities. The sensor data must be linked to BIM or external database to obtain storage functionalities. An algorithm for data fusion is required to aggregate and make sense of several data generated from multiple sources. This requires a computational power. The computation functionality provides data integration, automated decision making and processing of information and queries from the network and application layers.

c. Application Layer: It provides BIM-based visualization of sensor data and its access to end users. It enables end-user to interact with the object or surroundings under consideration through digital interface linked to IoT and devices in the network physical layer.

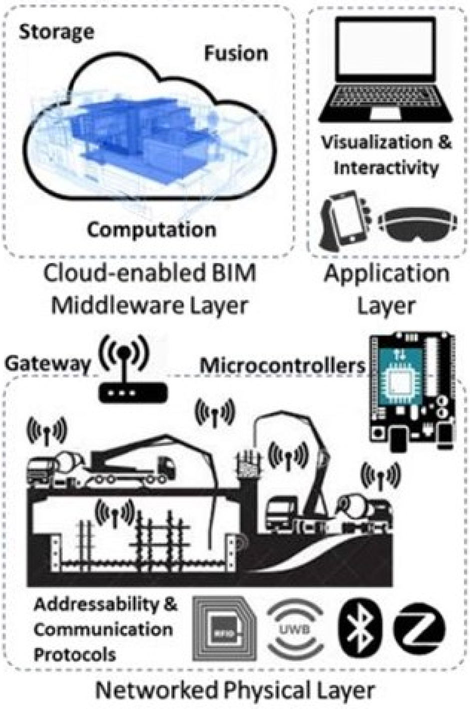

Storage
Fusion
Computation
Cloud-enabled BIM Middleware Layer
Visualization & Interactivity
Application Layer
Gateway
Microcontrollers
Addressability & Communication Protocols
RFID
UWB
Networked Physical Layer

Figure 2 Conceptual framework of generic BIM centred IoT architecture [20].

Similarly, authors in [21] provide a framework for integrating BIM and IoT. It is based upon development of platform that integrates building information using IFC with wireless sensor nodes through open API. It consists of three basic blocks as described below. The framework is shown in Figure 3.

a. IoT devices: Devices that gathers environmental data of the premises where they are installed.

b. Backend Server: It consist of loosely coupled IoT and BIM services that can communicate with each other following SOA (Service Oriented Architecture) design style. Standardized interface of O-MI (Open Messaging interface) and OD-F(Open data format) developed by The Open Group (TOG) in 2014 is used to access services of Backend server components. O-MI is used for transmitting life cycle related information mainly intended for automated processing by information system. OD-F can be used for publishing available data using URL addresses and when used together with O-MI it can be used for requesting and sending published data between system. It represents information about things in a standardize way that can be exchanged by all the information system that manage IoT related data. The backend function is to harmonize publishing and consumption of data through standardized Web-API for which OM-I and OD-F are used. This layer also provide functionality for interpretation of built environment data through IFC standard formats that can be either directly stored local data or in servers and accessed through APIs. The IFC file is further translated to web document using open shell platform and instantreality aopt tools for web-based visualizations.

c. Front end: Front end acts as interface between the user and backend services providing data in human readable forms. It can be implemented as websites, web apps or smartphone apps. It also takes queries from the backend services.

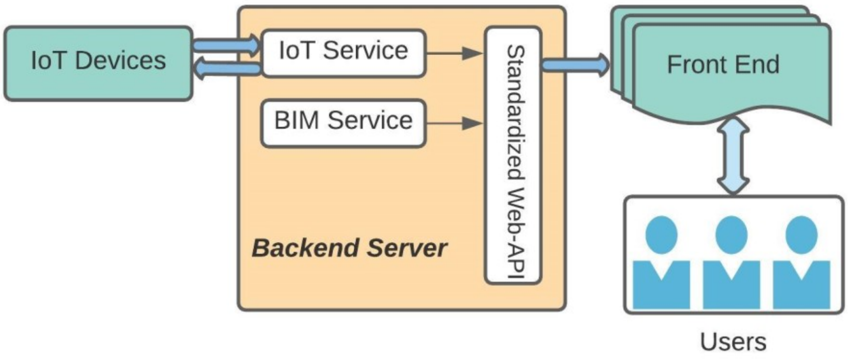


Figure 3 Conceptual system architecture for integrating BIM & IoT [21].

The synergy between BIM and IoT has potential to be used in various fields. In article [22] authors presents review of research works within integrated BIM and IoT technology and categorize its use for several domains. These includes followings.

a. Construction and operation monitoring: Several aspects of construction & operation process such as onsite environment monitoring, resource monitoring and localization, performance, and progress monitoring along with enhanced communication and collaboration can be benefited by real time information sharing capabilities of the integrated system.

b. Health and safety (H&S) management: IoT devices can provide valuable data on environment monitoring and human activities. These data together with BIM visualization and information sharing capabilities can be used to determine associated safety risks for working personnel. It can be used for health and safety trainings and on-site monitoring of H&S.

c. Construction logistic and management: The integrated system can facilitate automation in prefabrication using advanced sensor technologies. IoT devices like RFID(Radio Frequency Identification) can be used to track manufacturing process, logistic and its visualization in BIM model. This ensures reliable information flow and communication between humans, devices, and system. This timely information exchange across the supply chain adds value to the system.

d. Facility management: The process of operation and maintenance of building equipment, monitoring of energy requirement and performance management can be automated using the Integrated system. IoT devices along with visualization techniques can help in fault tracking, location tracing and access to real time energy data.

1.4.2 Related works

This section presents review of articles that is based on integrated frameworks of BIM & IoT. The review is mainly focused to tools and techniques that has been used in each of the articles.

1.4.2.1 Localization & Safety Management

In [23] authors introduce IoT approach that integrates environmental and localization data in BIM based platform to keep records of actual data sets. The data generated from the physical environment are stored in the cloud platform and interfaced to BIM softwares for visualizations. The environmental data is measured with Avent's Visible Things Bluetooth smart sensors. It is equipped to the helmets of the workers for collection of illumination data. For location tracking Accent systems iBKS105 BLE (Bluetooth Low Energy) beacon using Eddystone protocol and RSSI (Received Signal Strength Indicator) measurement for distance estimation is implemented. BLE are Bluetooth devices that works at standard frequency range of 2.4GHz and consumes low energy with reduced transmission power and shorter data transfer. These devices are supported by system compatible to Bluetooth 4.0. BLE device sends advertising events containing identification numbers, sensor data or short web address and waits for connection requests from smart devices. Eddystone is a Google's standard for BLE beacon communication protocol. RSSI determines the signal strength of the connected device for location tracking and theoretical distance can be approximated using path loss model [24]. In the literature, personal mobile device (smart phone) is used for capturing of data and sharing to cloud platform. The data for location tracking was stored and analyzed using IoT platform of ThingWorx while Microsoft Azure platform was used for storing environment related data. For visualization of environmental data authors designed dynamo script in Revit that would process real-time data from the Azure database. For location tracking BLE beacons installed inside the room (wall, ceiling, or door) transmit their signal and as soon as the signals were caught up by the smartphones of the worker inside the room, data were transferred to cloud-based platform via smartphones. The results of the set ups indicates potential for implementation of digital and smart technologies for data collection and visualization in construction industry and facility management and operation process. However, the article suggests for need of test in realistic construction environment and do not provide sufficient details on the integration of localization data with BIM platforms.

Similarly, in [25] author investigates BIM & cloud enabled RFID indoor localization system, consisting following tools

a. Two RFID readers (Astra-NA & Mercury 6) strategically placed around the sensing area to read tags on the object for its localization,
b. Three different types of RFID antenna to read data from tag and transmit it to the computer via reader,
c. Passive tags (using radio energy emitted by readers as power source)
d. Wi-Fi router for transmission of data from the readers to computer and cloud server.
e. Computer with monitoring and alert program for configuration and calibration of the system.

The tag information is collected by RFID antennas through radio waves. This raw sensing data includes tag number, entry time stamp, RSSI, antenna Id etc. RSSI value is used to locate the tag from the antenna based upon its proximity, for which lower RSSI value indicates farther from antenna. The location of antenna is provided to BIM database and 3D BIM model in visualization application. It is integrated to cloud computing system that provides contextual visualization. A dedicated program on the cloud server process and links the location data with project database. The program for real-time data monitoring was developed in visual studio. The cloud service is provided as SAAS, for which setting up the program and the data based on the cloud server is done in preliminary stage of the project. Here the private cloud service is used which is established on Georgia tech domain. This made it possible for visualization and access information remotely using web browsers. The field test of the system showed that it was capable to locate the subjects in real time with 88.1% accuracy, 84.7% precision and 89.6% efficiency. Some of the limitations of the system are mentioned as coverage of RFID antennas, system latency dependent to RFID readers, signal strength of Wi-Fi networks and slower refresh rate of RFID system.

In [26] the integrated system is used for fire prevention and disaster relief system. It makes use of fire sensors based on Bluetooth to provide information on temperature, smoke conditions and personnel locations in an indoor environment. The information is integrated to BIM based system for 3D visualization. It can be beneficial for early detection of fire, its point of origin and spread. The location data obtained can be used to provide guidance to personnel inside the building to support evacuation and rescue works. The personnel localization data are accessed using mobile app. The app makes connection to the fire sensors via Bluetooth and captures its broadcast information such as RSSI, sensor Id etc. These signals are uploaded to the server and are used to determine the location of the sensor or the mobile device based upon corresponding strongest RSSI, sensor Id in BIM component, and mobile device Id. The BIM model for visualization of environmental data and personnel location is obtained by using Autodesk Revit software and its API. The model includes 5 modules developed in windows operating environment (Windows 7) and .NET Framework 4.5 using C# programming. For database management and communication between cloud service and Revit, SQL Server 2010 is used. The sensors configurations, daily data collection and exchange with BIM/mobile app is intermediated using Raspberry Pi in the system.

1.4.2.2 Facility Management & Environment Monitoring

Location tracking for control of electric appliances using BIM virtual model is presented in [27]. Authors have developed a system where virtual environment of BIM model receives position and orientation of real-world handheld device used for control of electric appliance. The virtual environment is created using unity game engine and UWB (Ultra-wide band) technology is used for localization. It consists of 4 UWB anchors and 1 tag. The tag is placed to the device and the anchors are mounted on the walls of the room. The time difference between sent and received UWB signal is used to calculate the linear distance between the anchors and the tags. Location of the anchors is predetermined and are registered in the BIM model. One of the anchors is considered as an origin and all the X,Y,Z axis are defined for the room. This enables the position measurement of remaining anchors based upon the origin. The appliance controlling the device is equipped with microcontroller. The coordinates of each anchors are inserted to the software code in microcontroller written in processing language. These coordinates are utilized by the software program to determine the position of the tag. This positioning algorithm treats the anchors as reference frames and uses the set point to determine the coordinates of the device.

IoT devices based upon LoRa (Long Range) communication technology has been used in [28], for measurement of Temperature, humidity, and illuminance level of an indoor environment. These data are then visualized using an interactive 3D BIM platform and can be accessed through Smart devices (Mobile / Tablets). Unity game engine has been used to integrate BIM with IoT system to represent real-time data in the model. Using unity game engine platform, data can be visualized with various colors, avatars, and warnings. C# scripts are used to integrate BIM model with unity game engine and SQL language has been used to manage and access the database. The unity game engine platform presents BIM model with all the processed data and allow application deployment to various devices like web, mobile or windows. The communication between the IoT devices and application server is mediated by LoRa Server (Network Layer). The application server is responsible for collecting of data from the network server and processing. For, this Raspberry Pi is configured to run as full-time web server using Flash Frameworks written in Python. Incoming HTTP messages with data packets from the devices via network layer are processed by the server to create corresponding objects according to sensor type and send them to MySQL database on the same Raspberry Pi. An auxiliary display for reading sensor data is also presented using Node-red software.

The conceptual system architecture for integration og BIM & IoT as shown in Figure 3 and discussed in 1.4.1 has been also implemented by authors In [21]. Using IoT devices environmental data is integrated to web-based system, Otaniemi3D developed by the authors. The integrated system provides information about energy usage, user comfort and occupancy through O-MI, OD-F and IFC models. The data collected from IoT devices are stored in the database. The standard interface of OM-I and OD-F are encapsulated through simple programs and are used for communication with IoT devices and available services in the backend (IoT, BIM, Booking, Wi-Fi location etc.). Static IFC files incorporating spatial relation with IoT devices are translated to WebGL & SVG formats using OpenShell and InstantReality aopt tools. To recover losses associated to IFCsensortype and IFCspace during translation of file, additional Python script has been used. The front end of the developed system provide access to data analytics, floor plans, heat maps, 3D Model and sensor locations. These functionalities provide the visualization of average environmental data, trends & correlations, and access to sensors readings in an interactive 3D model. The developed platform, Otaniemi3D is presented as an open framework that can be used to develop compatible applications by using its APIs.

To support facility management and decision making in article [29], authors present BIM based visualization system for IoT devices that are used to collect temperature and humidity data of a room. These data are processed for color visualization in BIM platform based upon the comfort level. The tools used for visualization of data are Dynamo (plugin for Revit), to automate the sensor data readings, Arduino microcontroller to collect sensor data and Firefly suite to create node linking Dynamo and Arduino, that forms basis for importing sensor data in Dynamo. Similarly, authors in [30] presents BIM based visualization platform for sensors used to measure temperature, humidity, and carbon dioxide level. The authors consider a university building, with an area of 12000 sq.m spread over five storeys. Each rooms of the building are equipped with advanced Temperature and CO_2 sensors and readings are stored in the university local server. Here three methods for integration of sensor data with BIM visualization platform is presented. In the first method data are integrated to Autodesk Revit software using Dynamo. Two dynamo scripts are used, one for exporting the Revit model room attributes to database template in MS Excel, where temperature and CO_2 value are to be inserted. The second dynamo script is used to import CO2 and Temperature value into the software. 2D visualization is obtained with color coding based on data for each room. Similarly, the next method includes data integration using function of Navisworks software. The sensor database is accessed using ODBC (open database connectivity) driver within 'DataTools' function in the software and SQL is used to integrate model room element in the software and the database. Another function available in Navisworks, 'Appearance Profiler' is used for custom appearance and color coding of the objects in the model based upon sensor data. The third method implements Navisworks API integration and requires development of add-in for Navisworks. The authors developed the plugins including functions of general model information, assets search of facility, visualization of sensor data of selected level and visualization of historical data for selected rooms. Data from sensors are used to create 2 databases in MS Excel. These include sensor readings for each interval and average readings for each day. These databases are stored in the local server. Database containing average daily sensor readings are used to visualize data for each room and levels at selected date. While historical data in the software are generated through integration of database that includes sensor readings for each interval. The sensor readings of selected room are collected from database at defined time interval. The plugin is also used to provide color coded visualizations of the rooms based upon sensor readings. The plugins are developed using Microsoft Visual Studio (2015) .NET Framework and Navisworks SDKs.

1.4.2.3 Building & Infrastructure Monitoring

Article [31] presents BIM and IoT integrated framework used to find specific components in a construction site and change its installation parameter. The latest database obtained after the change of the parameter is then shared to BIM model that would update it in real time. As a case study authors used a room model of a construction area, with electrical and mechanical equipment. These equipment are provided with RFID tag and are to be read by a mobile RFID reader equipped in a Tablet. Once the equipment is read and the installation status is changed the database is updated with new status which are then updated to Revit model of the room. First BIM database is exported to Microsoft Access with ODBC feature available in Revit and then it is copied to the cloud storage. Later Revit API has been used to import updated database from the cloud. The data base in the cloud are updated by software developed in C# Visual Studio .NET Framework. Two softwares are developed using the platform and are installed to the mobile reader and a stationary computer. The software used in mobile reader is used to change the parameter of the component which is then received by the computer and updated to the cloud database. The communication between the software and cloud is based on XML. One RFID antenna is installed on the entrance of the room so that reader sends and receives signal to the tags. This integrated framework shows potential of usage in job site for component tracking, resource management and progress monitoring.

Similarly, in [32] development of integrated system is proposed with functionality of tracking physical resources in construction site. The article is focused on tracking of metaling formworks used for cast-in-place concrete. The basic tools that are used for this purpose includes handheld RFID reader , tags (attached to form works), Antenna, Raspberry Pi as an interface to connect with cloud database and BIM model including all the elements under consideration along with their unique Id associated to physical system. The article demonstrates the potential of the framework to be used for obtaining physical information of the resources, maintenance data and records, location, and positioning in AEC sector. However, the author does not provide clear methodology on development of user interface containing BIM model for visualization and access by other stakeholders.

Article in [33] provides guidelines for linking RFID technology to BIM and discuss about the development trends and illustrates decision tree for considering appropriate RFID & BIM system based on utilization domain. It suggests use of cloud-based BIM and passive UHF (ultra-high frequency) RFID for construction-based projects. Further suggestion is made on use of cloud-based BIM. This includes commercially available alternatives like Autodesk BIM 360, Graphisoft BIM server or developing own cloud BIM by exporting BIM data into an open format (IFC) and rendering data using interactive 3D graphics like WebGL (Web Graphics Library). The article also provides insights on how to extend data schema of the BIM when properties of target object are not included during file translation, gateway and processing of data using standard API of BIM etc.

Based upon above review of the literatures it can be seen that most of the IoT devices used for localization and facility management are based upon Bluetooth, RFID, and ultra-wideband technologies. The storage and management of data can be done through both cloud & local storage databases. The development of an integrated system is however based upon contextual model. This enables the possibility for development of an integrated frameworks that are dependent to utilization domains and functionalities. For example, in indoor localization, RF technologies are preferred due to their accuracy compared to GPS and cellular technologies.

BIM and IoT tools can be used to develop a platform which can visualize the real-time localization and environment data in an indoor area.

1.5 Theory

1.5.1 Indoor Positioning

Positioning can be referred as an estimation of an object or a person locations. Development of such technology is based upon ability of devices capable for contextual awareness of the locality and surroundings. IoT devices have been emerging with potential for realization of contextual awareness of its surroundings and are envisioned for development of smart environment [34]. Location information obtained from the positioning devices are dependent to functionality and sensing techniques of the devices. These information obtained can be classified in to four types physical location, symbolic location, absolute location, and relative location [35]. Physical location provides precise position of something expressed along with its coordinates. While symbolic locations are used to define where in space something lies, for example in office area or in Laboratory. Physical location information can be linked to derive symbolic information. Example finding a location of closest printer from the physical location point. Absolute location information is obtained when more than one device uses shared reference grid for all located objects and gives the same readings of coordinate for objects at same locations. Relative location information is obtained when devices have its own frame of reference and provides information on the object location relative to self. The location tracking system itself can be categorized in to four aspects which includes, positioning, variable ranging, and device the division tree is shown in Figure 4 below [36].

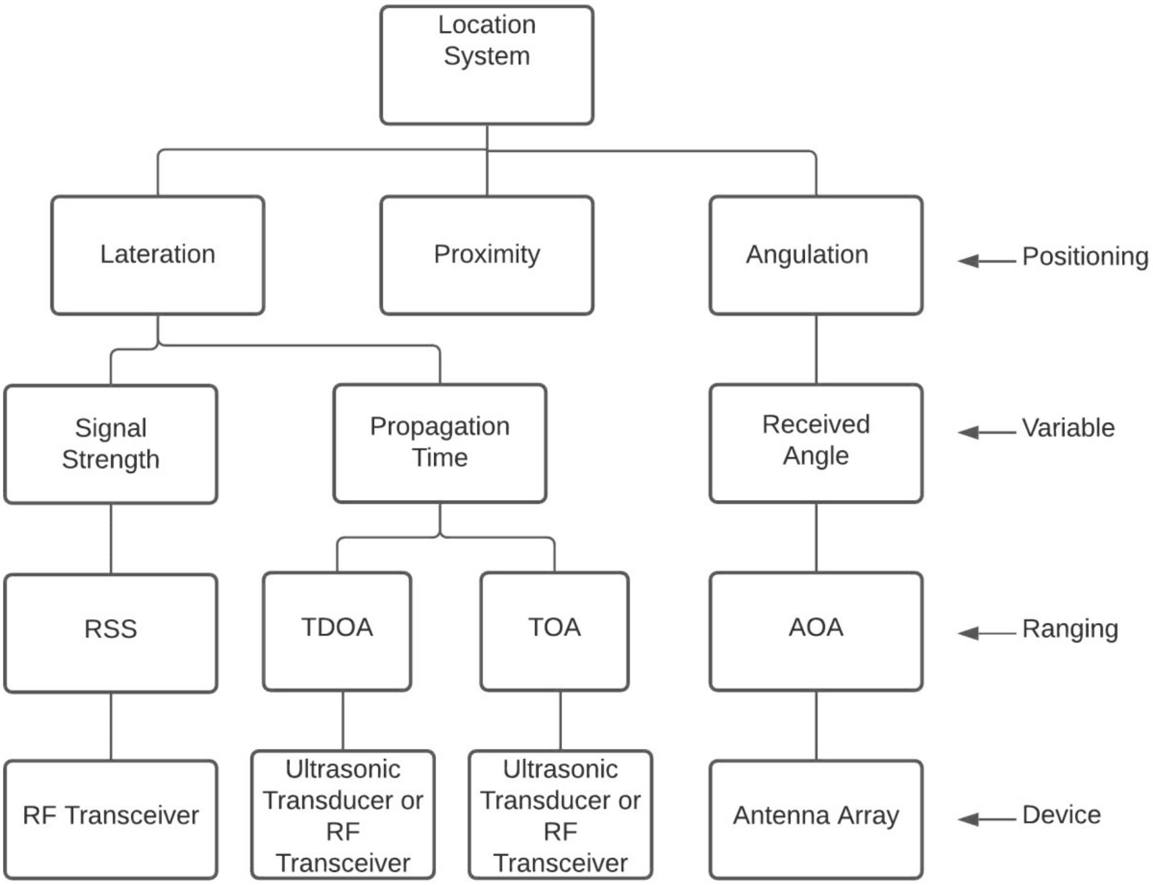


Figure 4 Classification of location tracking system [36].

Positioning in above categorization refers to techniques that are used to estimate the location of the target, through analysis of the variables. Variables can be obtained from sensor data. There are several algorithms that can be used to estimate the positioning for each of the methods categorized under ranging. Details on these methods and algorithms are mentioned in section 1.5.2. The device represents tools for the distance measurement. Antenna arrays are used for the measurement AOA (angle of arrival) [37]. RF Transceivers are used for RSS ranging method and have internal registers for storage of RSSI values. While RF and ultrasonic transceivers can be used for measurement of arrival time of the signals [38].

1.5.2 Measurement and positioning methods

Indoor localization is subjected to signal attenuation and distortions due to multipath fading and poor line of sight between the transmitter and receiver device. These are the major factors contributing to errors in measurement of locations [39, 40]. For mitigation of such measurement errors positioning algorithms are developed. These positioning algorithms can be divided in to three categories, Triangulation, Scene Analysis and Proximity [41].

1.5.2.1 Triangulation

In triangulation methods location of target object is estimated using geometric properties of triangles. It has two derivations lateration and angulation. Technique of measuring position of an object by measuring its distance from multiple reference points is known as lateration. The distance is derived using attenuation of the emitted signal strength or calculated from the product of radio signal velocity and time of travel. So, the distance information is obtained from Received signal strength indicators(RSSI), Time of Arrival (ToA) or time difference of arrival (TDoA). The angulation positioning method uses two separate positioning nodes with known distance to compute angle of arrival and locates object by computing angles relative to the nodes [41, 42].

A. Lateration

i. Time of Arrival: This method requires at least three reference station at known location to estimate the location of mobile target. It is based on the principle that distance from the mobile target to the reference stations is directly proportional to the propagation time. The coordinate of the target is calculated using one way propagation time and distance between the signal transmitter and measuring units. Therefore, it requires that all transmitter and measuring unit to be synchronized and each transmitting signal is labeled with time stamp. Once the exact time of the signal arriving at reference point and speed of the signal is known the distance from the reference points can be calculated using following equations [43, 44].

$$d = c * (t_{arrival} - t_{sent}) \tag{1}$$

$$d = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{2}$$

where c is the speed of light (speed of the signal) and , $(x_i, y_{i)}$.is the known position of the reference station. The possible locations of the target can be determined using distance calculated from equation (1) with equation (2). The set is to be calculated with enough reference points and requires at least three reference point to obtain 2D and four reference points for 3D coordinates of the target. The exact position is then obtained by finding the intersection as shown inFigure 5. There are also several other algorithms that has been analysed for ToA localizations. Some of these include, analytical method, Taylor series method, approximate likelihood method, least square methods, two-stage likely hood method, genetic algorithm etc.[45].

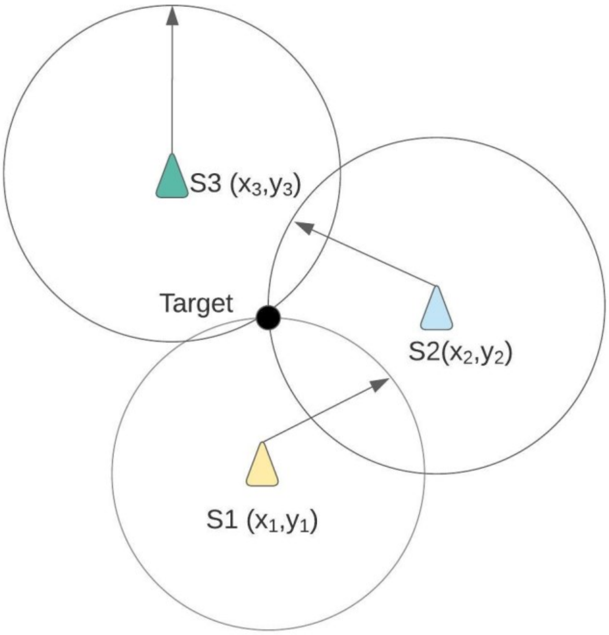


Figure 5 TOA based positioning measurement.

ii. Time difference of Arrival: Unlike ToA, this method does not require the record of time at which signal is transmitted from the target. But it is based upon the time at which the signal is received to the reference stations and speed of the signal. When time at which signal is received at two of the reference stations are recorded, difference between them can be used to calculate the differences in distance between the target and the stations. Following equations below can be used to calculate this difference in the distance and coordinates of target [44].

$$\Delta d = c * \Delta \mathrm{t} \qquad (3)$$

$$\Delta d = \sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2} \qquad (4)$$

Where, c is the speed of light (signal speed), Δt is the difference in arrival time of signal at two reference stations at positions (x_i, y_i) and (x_j, y_j) in 2D plane and (x, y) is the coordinate of the target. Using Equation (4), hyperboloid can be formed that includes possible location of the target. Each Δd would results in two branches however, with approximate location of target one of the branches can be discarded. Figure 6 below shows the possible location of the target using TDoA method.

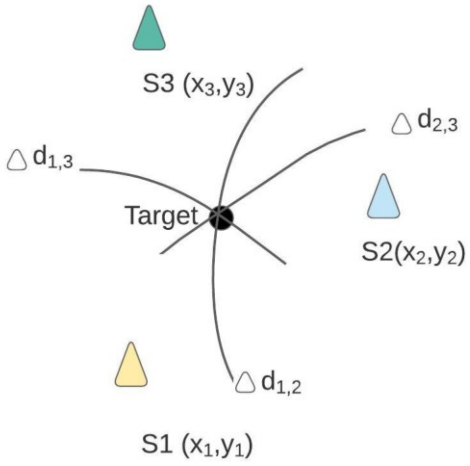


Figure 6 TDoA based positioning measurement.

TDoA and ToA methods are applicable where the target and the reference stations have a layout with clear line of sight. These methods assume that signals travels along the shortest path. So, whenever the path is blocked signal travel longer distance which results in incorrect target locations.

iii. RSSI based method: RSSI based method is applicable for the layouts where proper line of sight cannot be mentioned. It estimates the distance of the target based upon the attenuation of emitted signal strength. The widely used model for signal propagation is given by following equations.

$$I(d) = \frac{K}{d^a} I(0) \tag{5}$$

Where, $I(d)$ is the signal strength measured at distance d, K is the proportional constant, $I(0)$ is the signal strength at a unit distance, and a is the attenuation parameter of the signal propagation. The unknown distance d is calculated using measurement of $I(r)$ and parameters

a, K, $I(0)$ are modeled in advance. Following equation shows the traditional approach for calculation of distance d [46].

$$d = e^{\frac{\ln^{K} - \ln^{I(r)/I(0)}}{a}} \qquad (6)$$

However, these parameters are dependent to environmental conditions and often gives inaccurate readings with varying site conditions. So, an algorithm for environmental adaptive RSSI method as shown in [46] can significantly improve the location efficiency.

B. Angulation

In this method the location coordinate of target is computed from the angle between target location and reference location. The intersection of several pairs of angles lines formed by the circular radius from a reference station to the target gives location of the target as shown in Figure 7 [43]. It requires at least 2 reference points and 2 measured angles θ_1 & θ_2 to give location of target in 2D plane and 3 reference points for 3D plane. This method is used either with directional antenna or with array of antennas.

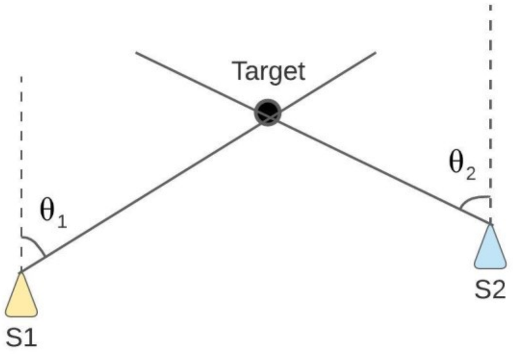


Figure 7 Angulation based positioning measurement.

The measuring units do not require time synchronization in this method. However, measurement using this method requires large & complex hardware, and angles measurement must be accurate in order to get the accurate position of the target [43].

1.5.2.2 Scene Analysis

In this positioning system, related information of a scene is first gathered, and the location is estimated based upon the information gathered. The target can be encoded with the information that would provide its symbolic location. such as, location information of the target encoded in a tag attached to it can be identified by its reader. Similarly, visual positioning system with computer vision technology are used to track objects in video data [47]. One of the common methods of scene analysis is RSS (Received signal strength) based location fingerprinting [43]. In location fingerprinting some characteristics of the signal are associated to the location . Initially the survey of the scene is performed, where location information associated to signal strength from nearby measuring stations are collected . These information are then matched with the target during the measurements. There are several positioning algorithms based on location fingerprinting that use pattern recognition techniques. These algorithms include probabilistic methods, k-nearest-neighbor (kNN),neural networks, support vector machine (SVM) and smallest M-vertex polygon (SMP).

1.5.2.3 Proximity

This method of positioning measurement provides symbolic location information of the target and are usually associated to measuring stations. In this method several stations are deployed to cover up an enclosed area. The location of each station is collected in a database. When a mobile target is detected by a single antenna, it is considered to be within the zone of that station. When more than one antenna detects the mobile target the receiving signal strength can be utilized to estimate location of the target with reference to stations, such that location of the station receiving strongest signals is allocated to the target. This method is simple implement and are often used with IR, RFID and BLE systems.

1.5.3 Indoor Positioning Technologies

The indoor positioning technologies can be classified in to four categories based upon the type of signal used for localization [48]. Radio frequency (RF) signals, Light (visible & infrared), Sound (audible &ultrasonic) and Magnetic field (natural & artificial) The devices used for each of these technologies can be divided in two categories of being an active and passive device. The devices that are capable of generating its own signal and dependent to external power source can be termed as an active device whereas devices independent of the external power source can be placed under passive devices. Some of the widely used localization technologies are mentioned and discussed below.

1.5.3.1 RFID

RFID stands for Radio frequency Identification. This system comprises of three main components, tag, reader, and antenna. Tags are electronic label that can be attached to any object and can be both active and passive. Readers are the devices that sends radio waves. These radio waves activate the tags when in range and decodes the data stored in the tag and transfer them to connected hosts. These data can include various information on tag Id and RSSI values etc. Antennas are devices that radiates and receives RF signals generated by reader and tags, respectively. The positioning is based on received data from tags which includes tag identifier, RSSI value and reading the location database of the stationary antenna/reader. RFID can be deployed in both indoor and outdoor environment. These devices provide discontinuous positioning and can be implemented for zone-based localization. Figure 8 below shows an equivalent diagram of RFID [49].

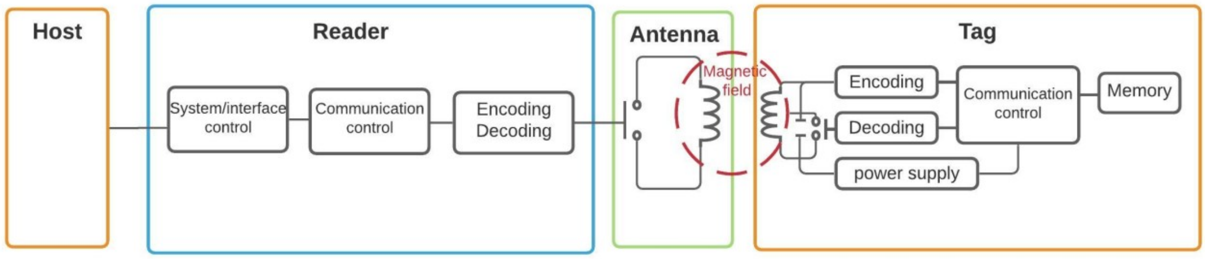


Figure 8 Equivalent diagram RFID system

These devices are widely used for access controls, equipment tracking, security and logistics [50]. These technologies can provide symbolic location information and can be used for the applications where accuracy can be compromised.

1.5.3.2 Bluetooth

Positioning and tracking of objects with Bluetooth is based upon BLE (Bluetooth low energy) beacons. These are devices that operates in 2.4-GHz ISM band and emit radio signals at predetermined intervals. The devices supporting Bluetooth (such as smart phones, tablets etc.) can establish connection with the beacons when in range. The beacons can transmit its identification data to the connected devices in two modes, advertising mode and connection mode. In advertising mode, beacons message (advertising) packets are sent regularly to the listening devices. The message packet hops between fixed sequence of three channels (37, 38, 39) that falls between the primary frequency thus, allowing advertising packets to avoid interference problems. However, mobile devices receiving these packets do not distinguish between these channels and combining their RSS values may often leads to reduced positioning accuracy. In connection mode data is transferred in a one-to-one connection [51, 52]. Data transferred by beacons are dependent to communication protocol used by beacons. Some common protocols for Bluetooth system are iBeacon and Eddystone protocols. Scene analysis method can be used together with BLE devices to gain more positioning accuracy. This would require an offline assessment of collecting reference data of the known positions along with their coordinates and beacon RSS values in these positions. Availability of such data in location database can later be mapped to the RSS value to estimate coordinates of the target by using positioning algorithms. BLE devices are portable, lightweight, and easily deployable. These devices consume low energy and are being used in many IoT applications. Challenges with BLE devices are their limited transmission power and range. Their RSS reading are susceptible to large fluctuations due to multipath fading, dynamic environments etc [52, 53].

1.5.3.3 Ultra-wide Band

It is RF signal with bandwidth greater than 500 Mhz or occupying portion of frequency spectrum greater than 20% of center frequency. This technology utilizes ToA or TDoA methods to estimate position of the target. An UWB system consist of tags and anchors. Tags send data request to the anchors and measures the response time from the anchors based upon which the distance is estimated. The positions of the anchors are predefined that helps in estimating target location using ToA or TDoA measures [48, 54]. This technology is considered to be suitable choice for critical positioning application due to its high accuracy, and precise measurements. These devices are comparatively immune to multipath fading compared to other devices and requires less power for extended operations [55, 56].

1.5.3.4 Zigbee

Zigbee for indoor positioning is based upon estimation of location using the received signal strength from the target and requires a system composed of network of sensors. These devices are equipped with radio transceivers and operates on IEEE 802.15.4 protocol using the frequency range of 2.4 GHz similar to Wi-Fi and Bluetooth. These devices are cheaper and are usually used for home automation purpose, where the lights and smart plugs are automated using Zigbee protocol. In comparison to Bluetooth and Wi-Fi, Zigbee have shorter range of 10-20 meters this makes it consume less power [57].

1.5.3.5 WLAN

Wireless local area network can be used for indoor localization and has a range of 50-100 m with gross bit rate of 11.54 or 100 Mbps [41]. Usually there are three approaches that has been commonly used for positioning using WLAN technology [48].

a. Propagation model of a known antenna can be used to calculate the distance to the known base.

b. Multilateration method is used based upon the relative strength of several known Wi-Fi stations.

c. Scene analysis, where the Wi-Fi stations with their relative strength is matched to a database associated with location.

1.5.4 Comparison of Technologies

The suitability of any technology is dependent upon application area and several factors that can be referred for evaluations of the technology. These factors can be termed as performance metrics and usually include the followings.

a. Accuracy: Accuracy in location system can be defined as an area surrounding the actual location within which percentage of repeated location measurement lie [58]. It defines the usual measurements of the system with possible errors in the distance from the true location. Usually mean distance errors are considered. Errors can also be represented as an offset from the actual position. Lower the distance error higher is the system accuracy.

b. Precision: It indicates the correctness of localization and is similar to accuracy. However, it is measurement of variation in the distance error over multiple trials [59]. Lower the variation in the error higher is the precision of the system.

c. Range: It describes coverage area of the system within which it can perform its functionality. Higher the range higher is the coverage area and less requirement of the devices. System with short range may requires additional devices to cover up the large area.

d. Complexity: It usually refers to hardware, software, and operational factors of the system. The software usually performs computational jobs. Time taken for computing the positions of the target is minimized if the positioning algorithms are performed on the server side [41]. The complexity increases if computational works are performed with softwares equipped with the mobile target devices.

e. Scalability: It refers to normal functionality of positioning system with increase in scope and size of the area and volumes to be covered by the system. Usually, scalability can be dependent upon the range of the system and with increase in scope of area to be covered, there may be requirement of additional devices. This may increase congestion in the wireless signals and data transfer.

f. Cost: Cost refers to many factors which includes, equipment, installation, maintenance, time, space, and energy. it may also relate to cost regarding computation for the system based on cloud computing and services.

Based upon some of the performance metrics discussed above Table 1 below provides comparison of indoor technologies.

Table 1Comparision table of Indoor positioning technology

TECHNOLOGY	ACCURACY [46, 58, 59]	POSITIONING TECHNIQUES [39, 46, 58]	RANGE [46, 58, 59]	COST [46]	POSITIVE [46, 60]	NEGATIVE [46, 60]
RFID	1.5-3.11 m	Proximity/RSS/ Scene Analysis	indoor	Medium	Low power consumption, Available in wide ranges	Low accuracy
BLUETOOTH	2-5 m	Proximity/RSS/ Scene Analysis	Building	Medium	High reception range, Low power consumption	Prone to noise , low localization accuracy
UWB	6-50 cm	Lateration (TDOA/TOA)	Indoor	High	High Accuracy immune to interference	High cost, Shorter range
ZIG BEE	25 cm	RSS	Indoor	Medium	Reusable infrastructure	Low precisions, requirement of special equipment.
WLAN	4.5 -6.5m	Proximity/RSS/Scene Analysis	Building	Low	Low-cost good precision	Vulnerable to access point changes

2 Methodology

A Room available in UiT Narvik, laboratory premises of integrated building technology was chosen to be modelled as an indoor environment for the case study with experimental set up. The room has an area of 5.4 X 4.2 m^2 and is often used by building department of the campus to carry out heating and ventilation related experiment. The room is equipped with lighting, heating, and ventilation equipment. For experimental setup in the room, first it was required to develop a 3D model of the room using BIM software, then an integrated system was to be developed to be implemented in the room for measurement of environment and localization data. These data were to be visualized in 3D model of the BIM software. Therefore, the scope of visualization was divided in to two parts. Facility Management and Localization. Temperature and illumination data were to be visualized as a facility management part. Whereas Localization data received using indoor positioning sensors were to be visualized in the localization part.

2.1 Development of Room 3D Model

The 3D model including geometrical parameters of the Room was developed using Autodesk Revit 2021. It is one of the popular BIM software in AEC industry and can also be used for development of new custom parametric objects and customization of predefined objects. It also provides easy user interface and supports range of applications [6]. The developed model in Revit was then exported to .Nwc format that is supported by Navisworks software and saved to .Nwd for visualization of data using Navisworks simulate software.

Navisworks is a BIM software that allows AEC professionals to combine several 3D models and detect for clash and collisions. It allows users to navigate around the model in real time, review them and disables the modification of 3D models to prevent any accidental changes. It is also possible to search and select the model items using Navisworks API, and develop a plugins for the software that can be used for the purpose of visualizations [30]. Figure 9 below shows 2D drawings available for the Test room which was used to develop 3D model in Revit. The exported 3D model of the room in Navisworks is shown in Figure 10 .

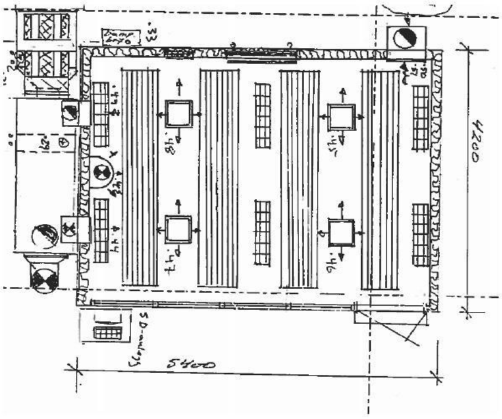


Figure 9. 2D available drawing of Test room

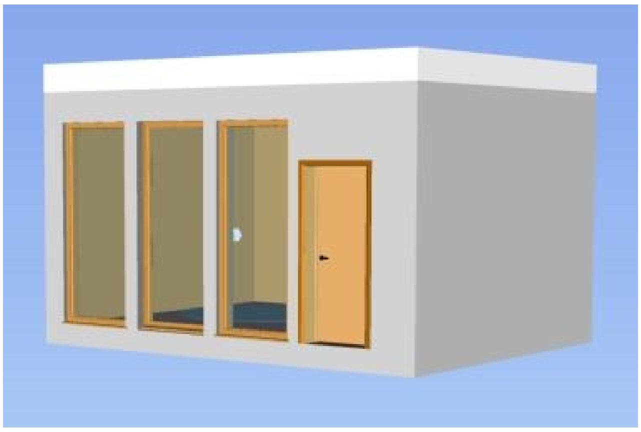


Figure 10 Developed & exported 3D model in Navisworks

2.2 Integration Framework

A detail literature review was carried out considering articles and works that has been conducted in the field of BIM and IoT integration. The reviewed articles are presented in section 1.4 of this report. Similarly, section 1.4.1 of this report presents potential frameworks that could be implemented in the project. It was decided to develop the integrated system based upon the framework presented in [20] and validate its usability in our case. This was found to be suitable approach as the framework is based upon several related works in the field. There are also other frameworks focused to web-based system development for data visualizations. Which Include translations of BIM models to web formats, reference implementations of IoT standards [21] or making use of Cloud based BIM technology. Implementing such frameworks at its nascent stage of development was prone to errors for our project and was considered to require more time and resources for experimental setups. Therefore, a new framework for our case study was developed based upon the conceptual frameworks presented in [20]. Figure 11 shows the developed framework.

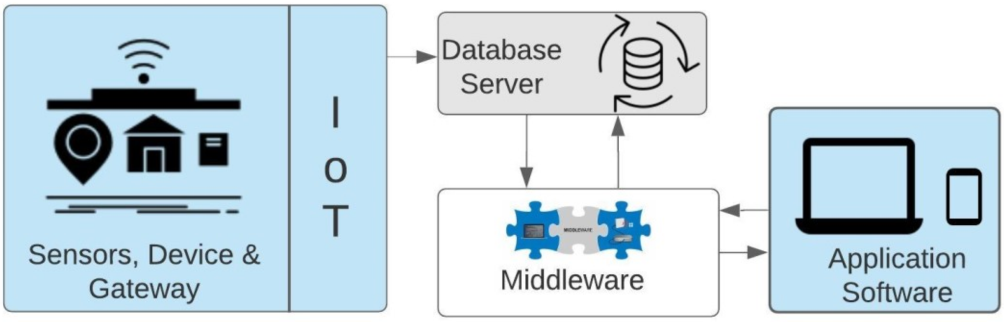


Figure 11 Developed BIM & IoT integration framework.

The developed framework consists of four Main blocks, IoT, Database, Middleware and Applications.

a. IoT block: It comprises of sensors and devices that is responsible for collection and filtering of physical data and transferring them to database server through gateway.
b. Database Server: It provides the storage and data management facility for received data from IoT block. It can also filter the data based upon the command received from the middleware blocks.
c. Middleware: Middleware is responsible for importing of data from the database server. Necessary commands are fetched to database through middleware for importing of data. Received data are integrated to Application for visualization using their APIs. It enables users to access data in database server through the application user interface.
d. Application: This block consist of BIM Software through which data are visualized. User can access to data through user interface of the software.
The case study included several tools that can be classified under each blocks of the developed framework. IoT devices and database server used for each part of the study were different. While same Middleware and Application platforms were utilized for both parts. Figure 12 illustrates Venn diagram for tools distribution in the case study.

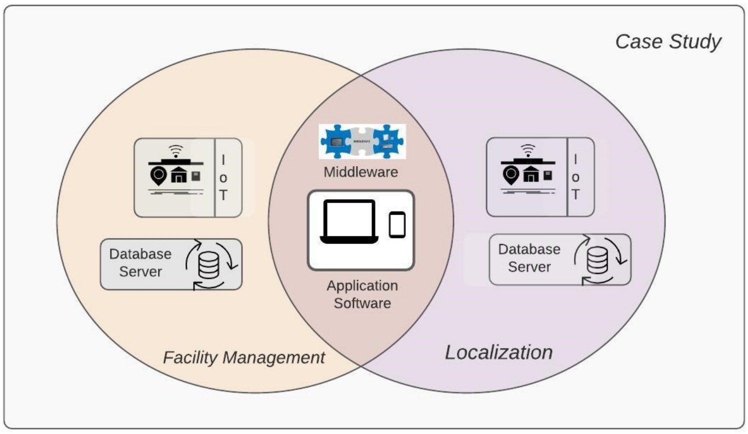


Figure 12 Tools distribution Venn diagram

2.3 Tools and Techniques:

Using the framework an integrated platform was to be developed that would visualize data received from the IoT devices. This included development of plugin for Navisworks platform that would support the visualization and data display functions.

2.3.1 Common Tools

As shown in Figure 12, Middleware and Applications were the common blocks used in both parts of the case study. The middleware block contains software that enables integration of IoT devices, data server and Application. BIM software platform to which data received from the sensors were to be visualized is referred as application.

2.3.1.1 Middleware

Microsoft visual studio 2019 was used for development of the framework in the form of plugins that is supported by Navisworks simulate 2021. The developed plugin was to be integrated with the Navisworks (NW) software to access sensor data. NW 2021 supports .NET Framework version of 4.7. So, C# was used to write a program in .NET framework with Visual studio 2019. Connection to database were made Importing libraries based on data access APIs and using connection string of the database. Navisworks API with related libraries were used to support data exchange and visualizations through developed plugin of Navisworks .

2.3.1.2 Application

Autodesk Navisworks Simulate 2021 was used as BIM visualization platform for this case study. A plugin developed in VS 2019 along with Navisworks API provided base for the data visualization for the case study.

The family objects and elements that are used to develop 3D model in Revit can easily be accessed using ‘selection tree’ window in Navisworks. The objects are represented as model items in the selection tree and can be identified with unique element Id values associated to its properties. This value can be passed using Navisworks API to select particular model item and override its color and transparency. This method of overriding the colors of model items was used to provide visualizations based on the data values received from the sensors. Thus, development of plugins to display the sensors readings and applying corresponding changes to the model item was essential for the case study.

2.3.2 Facility Management Part

For the measurement of environmental data, temperature sensor MCP 9808 and illumination sensor of model BH1750 was considered. The sensors are based on I2C (Inter-integrated chip) communication protocol. In I2C protocol data is transferred in form of messages that can be broken up in to frames of data. Each of this frame contains binary address of the slave and data frames that contain data being transferred. The message also consists of start and stop conditions, read/write bits, and ACK/NACK bits between each data frame [60]. Raspberry Pi 4 was used as master to control and read data from the sensors (slaves).

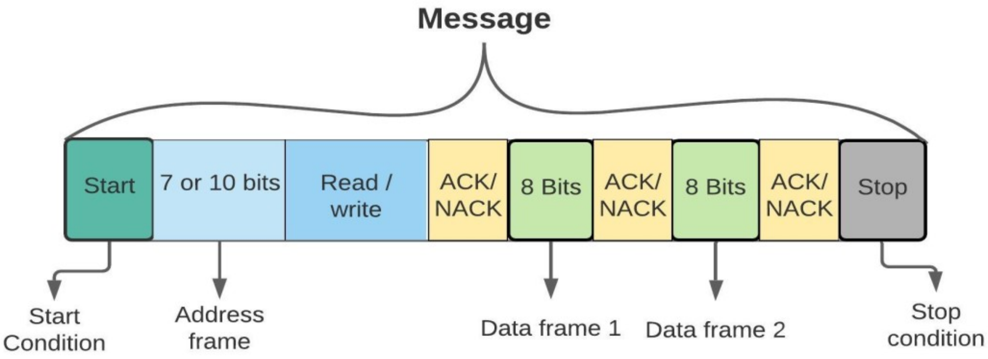


Figure 13 I2C Message breakdown [60]

2.3.2.1 IoT devices

a. MCP 9808: It is a digital temperature sensor that can measure temperature between -40^0C to 125^0C with typical accuracy og ±0.25^0C and precision og +0.0625^0 . This sensor needs to be connected with microcontroller device for reading of data. The sensor communicates with the device using I2C standard.

b. BH 1750: It is a digital ambient Light sensor for I2C bus interface. It is suitable for obtaining of ambient light data as it can measure wide range of illumination at high resolutions (1-65535 lux). The sensor gives its reading in Lux, which is the unit of illumination and is measure of luminous flux per unit area.

c. Raspberry Pi: It was used to read data from the sensors and export them to database server. For this the sensors were connected to the Pi module and python script running in the module would constantly read the data from the sensors then it was passed to the database server.

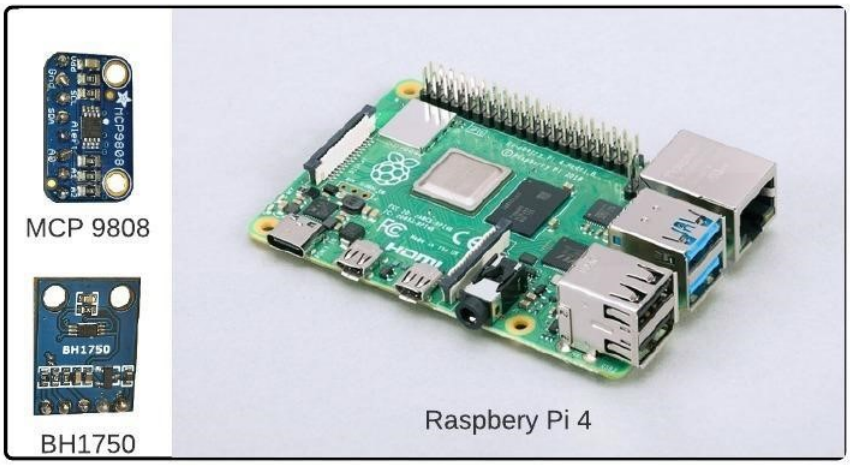


Figure 14 IoT devices used in Facility Management Part

2.3.2.2 Database server

For the storage of data, Microsoft Azure cloud service was considered in this part. Microsoft Azure provides SQL server service in cloud platform as fully managed database engine in the form of PaaS. It handles most of the database management functions such as upgrading, patching, backups, and monitoring without user involvement and enables creation of data storage layer for IoT applications in Azure cloud service. Database Query editor was then used to place query and access data from Azure SQL server database.

To store the data in Azure SQL database from IoT devices it would require two more cloud services to be used simultaneously. These services are IoT Hub and Stream Analytics services and are available in Azure platform.

a. IoT Hub: It is a PaaS service available in azure, that allows bidirectional communication between the cloud and IoT devices. It enables monitoring of data, gaining insights on IoT device and develop custom application for IoT platform. IoT Hub allows user to register IoT device, once the device is registered a connection string is generated that can be used in the IoT device to connect it to the IoT Hub. This enables sharing of data from IoT device to IoT Hub, through the connection string [61].

b. Stream Analytics: It is real-time analytics and event-processing engine that can be used to analyze and process fast streaming data from multiple sources simultaneously [62]. It is a helpful tool that can be used for analyzing real-time telemetry streams from IoT devices. It consists of input source, query, and output. As an input source to stream analytics several services like Azure Event Hub, Azure IoT Hub or Blob Storage can be used to ingest data.

Query is based on SQL query language and can also be used for data filtering. The output side of the stream analytics data can be assigned to storage services available in azure or can be used to provide alerts, or real-time visualization through services like Power Bi etc. that are available in the cloud platform.
In this case study the IoT hub was defined as an input source and SQL database was defined as output for the stream analytics. The query command included selecting of data from the input source and placing it to the output. No data filters commands were used in the query.

Figure 15 below shows the process map for facility management parts and overview of the tools used.

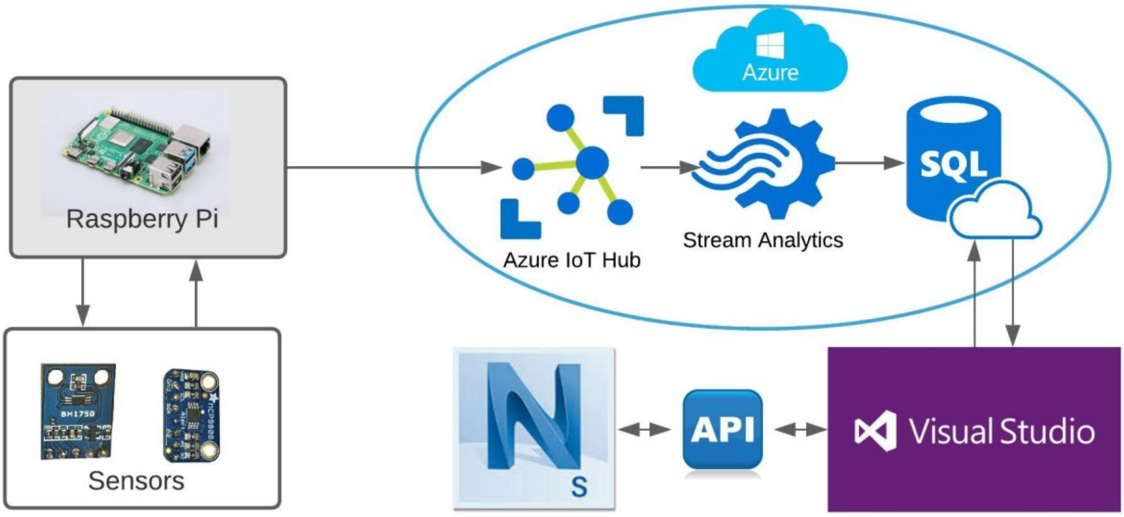


Figure 15 Process map and Tools used for Facility Management.

2.3.2.3 Plugin development and Visualization approach

This part included two main programs that ran simultaneously for measurements of sensor data and processing these for display in the application software. The IoT devices were controlled by a python program written in Raspberry Pi module for measurement of data and sending it to cloud storage in real-time. Similarly, second program was for plugin development and should be able to access data from the cloud storage using Navisworks platform simultaneously.

a. Method for data generation and export: The Python program written in Raspberry Pi 4 module consisted of several libraries such as System Management Bus (SMBus) that supported I2C connection, asyncio for supporting database connection and distributed task, time to access timer function, json for data formatting, IoT Hub device client to make connection with azure

IoT hub & datetime to place time stamp on recorded data. The program utilized timer function to get the data from the sensor at predefined interval. Then the data were transferred to IoT Hub, utilizing the connection string that was available during the registration new IoT device in IoT Hub. The data were transferred to IoT Hub in JSON format and included Room id, Time stamp, temperature in ^{0}C, and illumination reading in Lux. Element id value of the room defined in the 3D model was assigned as Room id. The data transferred to IoT Hub, was passed forward to azure SQL server database table using stream analytics in real-time.

General configurations, required libraries , Raspberry Pi connection diagrams to sensors and python script written for collection of sensors data and is export to azure IoT Hub, is attached to Appendix A of this report.

b. Data import and Visualization method :The program for development of plugins and visualization of data using C# .NET Framework in VS 2019 included several libraries related to ribbon development, Navisworks API, SQL server etc. Firstly, a ribbon control was created, along with button icon, and additional user controls were added in a tool panel with respective tab names for data visualization. The azure SQL server database was accessed using its connection string and query was placed to select the latest data reading from the database table. This would provide latest data with Room id, time stamps, temperature, and illumination readings. This functionality was inherited to 'Show Readings' button of the user control dedicated for facility management part. So, whenever the user clicks the button, the query is placed, and data could be accessed. However, this would require parallel running of program for sensor readings and azure Stream analytics, to place data in SQL server database table.

Using Navisworks API, value of Room id, was used to search and select the room in the model and corresponding temperature value was checked such that if the value of temperature was above 24^0 C the room colour would turn to red, if it was between 20^0 C to 24^0 C the room colour would turn to green and if it was below, 20^0 C then the room colour would change to blue. Beside this all-other model items were turned transparent for clear colour-coded visualizations of the room as per the temperature level. The colour visualization would also define the comfort label as information on the tool panel. Such that it would show comfort label to be 'Hot', 'Comfort' or 'Cold' based on temperatures level of above 24^{0}C, between 20^0 to 24^{0}C and below 20^0 C respectively.

The developed tool panel for facility management and the C# program written to access data from azure SQL server database for colour-based visualization in Navisworks platform is attached to Appendix B of this report.

2.3.3 Localization Part

2.3.3.1 Indoor positioning (IP) device:

The selection of indoor positioning device for localization part was based upon several parameters considered in section 1.5.4. Bluetooth based technology was found to be suitable for the purpose of our case study due to less complexity in software and hardware parts. Bluetooth device poses no interference problems to other devices nearby. It offers low installation and maintenance cost and are easily available. Similarly, there is wide use of Bluetooth integrated technology that are easily accessible and can be used for future research purpose along with localization system.

This case study used BluEpyc BLE devices for indoor localizations. The BLE devices consist of three main components. Reader/Gateway, Echobeacons and beacons.

a. Reader/Gateway: The model of the Gateway used is BE-BLEG-WE5.1. It is an indoor wall mounted gateway that supports Bluetooth 5.1 and is integrated with 1 antenna. It has functionality of both gateway and reader i.e., it is capable to read data from other BLE devices (beacons and echobeacons) and transmit the collected data to a Host (Connected PC) using TCP/IP protocol. It has read range up to 100 meters with transmission power of +3bdm and has receiving sensitivity of -85 dbm to -91dbm. It requires an external 12 to 24V DC power source.
b. Echobeacons (EB): Echo beacons are BLE devices that scans for the beacons within an area and allows identification and localization of the beacons. The echo beacons receive data from the beacons and retransmit the collected beacons data to the Gateway. The model of Echo beacons used is BE-EB5.WM.1. Each echobeacons have its unique address and support Bluetooth 5.1. Its transmission power can be varied from -20 dBm to +8 dBm. The transmission range is up to 200 meters. It has several configuration parameters and can be configured to filter the beacons based upon RSSI values, advertising masks, whitelists, and blacklists. It requires an external DC power source of 12 to 24 V.
c. Beacons: These are BLE mobile devices with unique address that' sends advertising data with limited energy consumption. The rate of advertising interval and transmission power can be configured using 'LightBlue', mobile App. The transmission distance is up to 120 meter and

has 4 level of transmission power that can be selected (-23dB, -6dB, 0dB and 4dB). It utilizes 3V dc power source and weight about 4.2 g with size of 31mm in diameter a 10 mm thickness. These devices can track location of the people and assets based on proximity method.

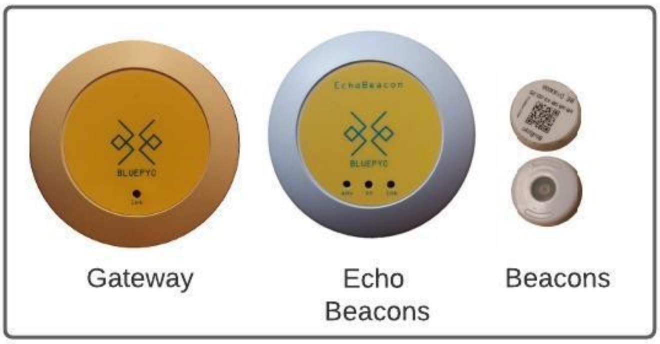


Figure 16 BLE Indoor Positioning devices

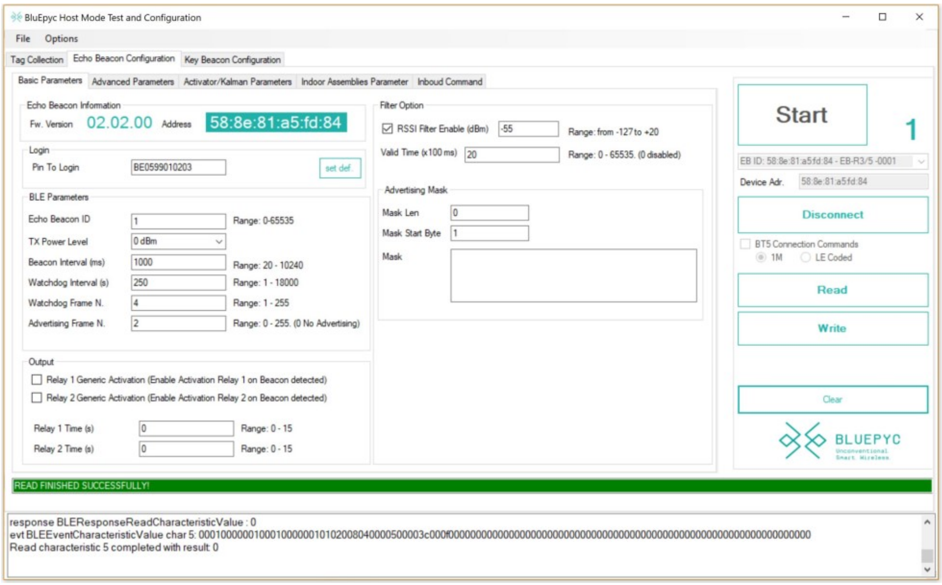
Table 2 provides an overview on the models and numbers of BluEpyc BLE devices that are used in the case study for localization part.

Table 2 Overview of indoor positioning devices used for case study.

BLUEPY BLE DEVICES	MODEL	QUANTITY
GATEWAY	BE-BLEG-WE5.1	1
ECHO BEACONS	BE-EB5.WM.1	2
BEACONS	Disk beacons	4

2.3.3.2 Configuration and functionality of IP devices:

Indoor positioning devices must be configured correctly to cover the area of their placement. Gateway and Echobeacons were configured using manufacturer's software, BluEpycStart. An example of Echobeacon configuration is shown in Figure 17 .Whereas Beacons were configured using the LightBlue app. Some of the Basic parameters considered for the configuration of these BLE devices are described below:



BluEpyc Host Mode Test and Configuration
File
Options
Tag Collection
Echo Beacon Configuration
Key Beacon Configuration
Basic Parameters
Advanced Parameters
Activator/Kalman Parameters
Indoor Assemblies Parameter
Inboud Command
Start
1
Disconnect
Read
Write
Clear
BLUEPYC
Unconventional Smart Wireless

Echo Beacon Information

Fw. Version 02.02.00 Address 58:8e:81:a5:fd:84

Login

Pin To Login: BE0599010203 — set def.

BLE Parameters

Echo Beacon ID	1	Range: 0-65535
TX Power Level	0 dBm	
Beacon Interval (ms)	1000	Range: 20 - 10240
Watchdog Interval (s)	250	Range: 1 - 18000
Watchdog Frame N.	4	Range: 1 - 255
Advertising Frame N.	2	Range: 0 - 255. (0 No Advertising)

Output

□ Relay 1 Generic Activation (Enable Activation Relay 1 on Beacon detected)
□ Relay 2 Generic Activation (Enable Activation Relay 2 on Beacon detected)

Relay 1 Time (s)	0	Range: 0 - 15
Relay 2 Time (s)	0	Range: 0 - 15

Filter Option

☑ RSSI Filter Enable (dBm)	-55	Range: from -127 to +20
Valid Time (x100 ms)	20	Range: 0 - 65535. (0 disabled)

Advertising Mask

Mask Len: 0
Mask Start Byte: 1
Mask:

EB ID: 58:8e:81:a5:fd:84 - EB-R3/5 -0001
Device Adr. 58:8e:81:a5:fd:84

□ BT5 Connection Commands
● 1M ○ LE Coded

READ FINISHED SUCCESSFULLY!

```
response BLEResponseReadCharacteristicValue : 0
evt BLEEventCharacteristicValue char 5: 0001000000100010000000101020080400005000003c000f00000000000000000000000000000000000000000000000000000000000000
Read characteristic 5 completed with result 0
```

Figure 17 Configuration of Echobeacons

A. Gateway settings: In this case study the gateway functionality was to transfer the data received from the echobeacons to the connected host (PC) in csv format. The gateway could also read advertising from other Bluetooth devices beside echobeacons and beacons. So, it was necessary get advertising packets only from the echobeacons. For this Mask parameter was used to filter other Bluetooth devices in the gateway. Also, the modes of gate way operation and scan interval were configured. Details on each of these parameters are mentioned below:

i. Mode of operation: The gateway operates in two modes, Host, and Notification mode. In Host mode it waits for commands from host and answers to each command. In Notification mode the gate way continuously asks to connected reader for transponders (beacons/ Echobeacons) in the RF field. If reader responds with one or more transponders in the field the gateway inserts them in internal queue and send them via TCP/IP using the notification format to the host. The host receives a row of data each time. For the case study notification mode was selected to generate csv data from the gateway reader in the host.

ii. Scan Interval: It can be defined as the time interval from when the device started its last scan until it begins the new subsequent scan. During one scan interval period the reader scan on one channel and all the three channels are used by the reader, respectively. The range of the interval varies from 3ms to 10240ms. This parameter was used to set the scan interval for the reader in notification mode. It was kept to 500ms.

iii. Scan Window: It is the time duration for Gateway reader to listen for the advertising message from other BLE devices. Scan window lies within the scan interval and can be less or equal to scan interval. If the scan window is less than the scan interval, reader will listen for advertising message only for the specified duration of the scan window and waits for remaining duration of the interval to be elapsed. It was used to define the scan window for the reader. Its value was kept to 500ms.

iv. Mask: This parameter was used to set the mask for advertising received from the BLE devices. If the Mask matches the Advertising the readers accepts the message otherwise it is neglected. This would help in filtering of BLE devices. The mask was placed such that only the Echo beacons could be detected by the gateway.

B. Echo beacons settings: The main functionality of EBs for this case study was to receive the advertising packets from the beacons and send it to the gateway. It could get advertising packets from all available Bluetooth devices in the RF field. Therefore, it was necessary to configure the echobeacons such that only the advertising from selected beacons is received. Some of the parameter considered for configuration of these filters along with scan and transmission power are mentioned below:

i. Transmission Power: It refers to the power level that is used by the device to send the advertising packets received from the beacons to Gateway. It was kept to 0 dBm for our case study.

ii. RSSI filter: This filter was used in EBs to enable only the packets received with a power level over the threshold defined by filter. This would filter the beacons out of the RSSI filter range. The RSSI filter value was set to -55 dB.

iii. Beacon List Filter: This filter consists of list of Beacon address that can be loaded and stored on echobeacons flash memory. The list can be enabled as 'Whitelist' or 'Blacklist' for advertising. This parameter was used to write 'Whitelist' of the beacons with their addresses such that only advertising received from the beacons in the whitelists are processed. Advertising received from any other devices outside of whitelist are filtered out. The whitelist was added containing 4 beacons addresses that were to be used as target in experiment.

iv. Watch Dog interval: This parameter defines the interval of time that Echo beacons waits before sending new watch dog packets when no beacons are detected. Watch dog packets are sent by the Echobeacons to listen advertising form the beacons. The range of watch dog interval varies from 1 sec to 18000 sec. It was set to 250 sec.

v. Watch Dog Frame Number: This defines the number of times the watch dog packets to be sent by Echobeacons when watch dog interval elapses. Its range varies from 1 to 255. It was kept to 4.

vi. Beacon interval: It is the time interval that is used by the Echobeacons to send the advertising packets received from the beacons to the gateway. The range varies from 20msec to 10240msec. Its value was kept to 1000ms.

vii. Advertising Frame Number: This defines the number of times the advertising of each detected beacons is sent to the gateway. Its range varies from 1 to 255. This was kept to 2.

viii. Valid Time: It is defined as the period of time within which, echo beacons does not send a beacon that has already been sent. After this time period if the beacon is detected again then the beacon is sent and new the new valid timer starts for the Beacon. Its setting range varies from 0 to 6553.5 sec, where placing 0 disables the feature. This value was kept to 2 sec.

C. Beacons configuration: The main functionality of the beacon is to advertise its packets so that Echobeacons can detect it. Therefore, it is necessary to configure the strengths of its transmission signals and interval of advertising. Following parameters are considered for the configuration of the beacons.

Broad casts interval: This defines the time interval from the beacons' last advertising packets to new advertising packets. Its range varies from 100msec to 10sec. it was kept to 250ms.

Measure and Transmission power: Measure and transmission power defines power level of the advertising packets transferred by the beacons. Higher the transmission power longer the reading distance. The range varies from 0 to 3 dBm. Where 0 defines low transmission power and short reading distance and 3 defines highest transmission power and long reading distance. The beacons were grouped in set of two, one set were to be used for personnel localization for which, the transmission power level was kept to 3dBm. While for remaining beacons transmission power level was kept to 2dBm.

2.3.3.3 Database server:

Unlike facility management part this part utilized Microsoft SQL server available in local PC for management and storage of data. The indoor localization gateway device collected localization data and saved it to .csv format in the host. In our context host is a PC to which the gateway was connected through ethernet. The csv data was then exported to SQL server database using Python script. The python script was used to filter the data and transfer it from .csv file to SQL server database tables. Once the data were arranged to database tables it was than ready to be picked up by the middleware blocks for visualization in the applications.

Figure 18 below shows the process map and overview of overall tools and softwares used for the localization part.

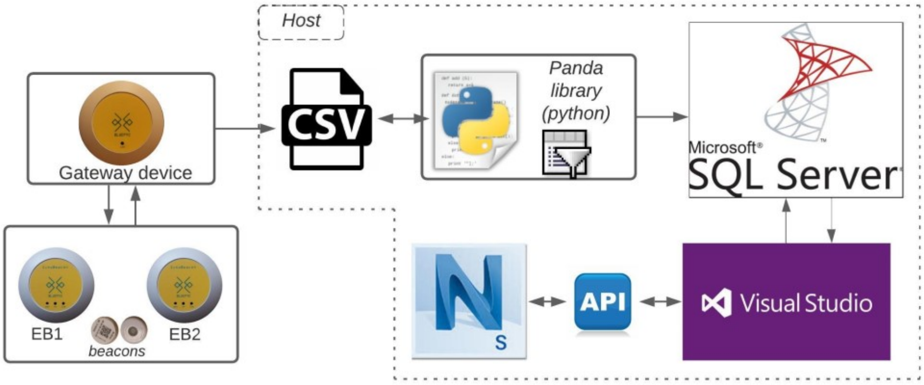


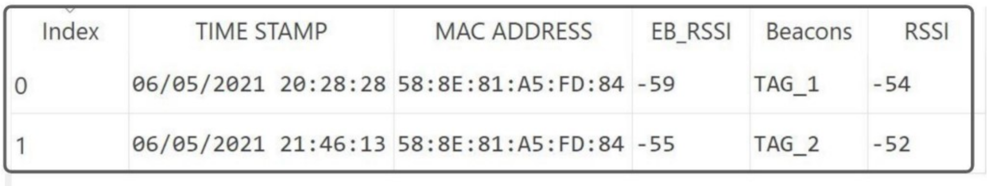
Figure 18 Process map and tools used for Localization part.

2.3.3.4 Plugin development and Visualization approach

Plugin development and visualization approach includes data filtration and export from indoor positioning devices to database server and importing it to middleware platform for visualization through the application software.

a. Method for data filtration & export : A separate python script with panda library was used to read csv file generated by the gateway and filter its data. Initially, the csv file included 8 columns of data. All of these data were not required so it was filtered to 5 columns of data. Based on the echobeacon address the data were then tabulated in to two tables, one for each of the echobeacons. Each of these tables included Time stamp, Echobeacons Mac Address, Echo beacon RSSI values, beacon address and beacon RSSI. For semantic naming of the beacons, its original address was replaced with Tag numbers in the program as shown in Figure 19. This data was than exported to SQL server database table in the same host computer using its connection string. The python program was written such that it would read the csv file in an interval of every 2 seconds and update the data table in the database with the new rows of readings. Considering if the new rows contain a beacon that was read by both Echobeacons the program would assign the beacon in the table of echobeacon receiving higher signal strength (RSSI value) from the beacon.

The python script to accomplish above mentioned tasks is attached to Appendix C of this report.



Index	TIME STAMP	MAC ADDRESS	EB_RSSI	Beacons	RSSI
0	06/05/2021 20:28:28	58:8E:81:A5:FD:84	-59	TAG_1	-54
1	06/05/2021 21:46:13	58:8E:81:A5:FD:84	-55	TAG_2	-52

Figure 19 Filtered data for export to database.

b. Data import and visualization method: Both the programs for facility management and localization were developed using .Net framework in visual studio 2019. Therefore, Additional tab for localization was created in the same tool panel of the plugin as discussed in section 2.3.2.3. In order to visualize the localization data, it was necessary to separate particular area for each echobeacons. So, the model of the test room was modified in Revit and included zone area with diameter of 1.5m allocated for each of the echobeacons. To represent echobeacons and gateway a custom object from electrical fixture family was created in Revit and zones were represented using object from the Revit floor family. Then 3D view of the model was exported to format that is supported by Navisworks. Figure 20 below shows the layout of echobeacons, zones and gateway in the test room model.

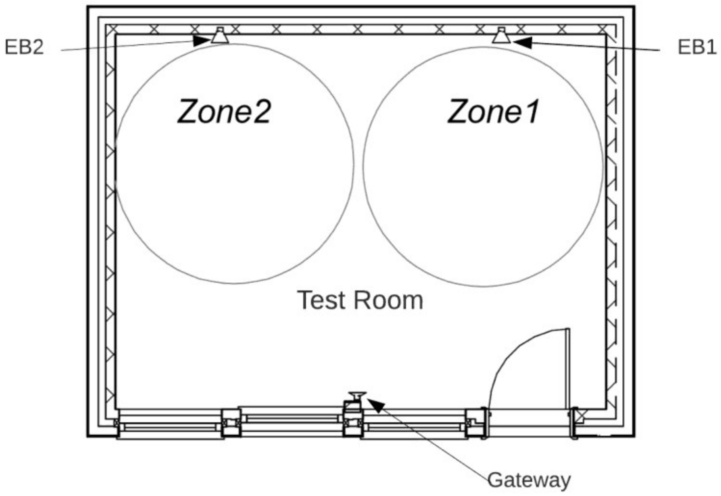


Figure 20 Indoor positioning device layout

As discussed in above section, the data to the database tables were passed using python program & csv file. It was updated, at an interval of 2 sec whenever a beacon was present in the zone or in the range of the echobeacons. The tables in the database were named as zone1 & zone 2, where zone1 represented data read by EB1 and zone2 represented data read by EB2. It was necessary that developed plugins connects to SQL server database & import the stored readings in the tables. SQL server database connection string was used to connect and import data. After opening the connection to SQL server database, SqlDataReader function was implemented in the program to import and read each row of data from each of the tables.

While executing the SQL reader function in the program each row of the data was passed as an argument to 'visualize' function of the program. Such that whenever data row from zone1 was passed to the function, element id value of the floor object that was named as Zone1 and custom object corresponding to echobeacon at Zone1(EB1) was searched using 'SearchCondition', of

Navisworks API. After finding these elements, their color was changed to green, to convey visual message that the beacon was present in that area. Similar approach was used for zone 2. Beside color coded visualization in the model the plugins would also visualize the data from each of the zones in the tool panel's text boxes that included the number of tags present in the zones along with their tag names. Also, two data grids, one for each of the zones were used to visualize overall imported data. All of these visualizations would work simultaneously when the data were read from the SQL server database.

The program would read SQL server data in every 1 sec for new updates, for this a timer function was used. The timer function could be enabled by pressing 'Start' button under localization tab of the tool panel. When the time interval was elapsed, it would trigger an event and call, 'ReadSQL_Data' function that would read newly updated data from the SQL server database. If there were no updates, then in user control a message would show 'no recent updates in Zones'. When no updates were available the color visualization were reset to original state and the text boxes display showing tag names and numbers were cleared.

The developed plugin contained 3 buttons 'Start', 'Stop' and 'Details' Pressing 'Stop' button in the plugin would stop the visualizations and backup the data from previous two tables to new two backup tables for redundancy. When 'Details' button was pressed a new form would appear. In this form it would be possible to get insights on number of tags present in the location along with their entry and exit time stamps. For this it would require users to fill the time periods and pick the zones of interest to access the data.

For real-time visualization of the localization data, it was necessary to run BluEpyc devices for data generation, Python scripts for data filtration and export and the developed plugin in Navisworks simulate application simultaneously .

The tool panels developed for Localization data visualizations and C# program for localization are attached to Appendix D of this report.

2.4 Experimental Set up

The developed integrated framework of BIM and IoT was set up to run in the test room. This included installation of the devices in the room for both facility management and localization part. The detail on each of these set ups are described sections below.

2.4.1 Facility Management:

The Sensors were placed in the breadboards & connected to Raspberry Pi module. The set up was placed inside the room and a python program was run in the Pi module that will get the temperature and illumination readings from the sensors at predefined interval of time in the program. For test purpose the interval was set to 30 sec. The variation on the temperature and illuminations were, made to get the visualized results. However, for color-based visualization the color parameters of the room were only made dependent to temperature variations.

The sequence of temperature readings taken were.

a. One reading greater than 24^0C.
b. One reading less than 20^0C.
c. One reading between 20^0C to 24^0 C.

Visualization results are shown in section 3.1.

2.4.2 Localization part

The experimental set up for indoor localization using Bluetooth devices is shown in Figure 21. Each of the echobeacons were placed at the height of 1.3m from the floor level facing parallel to each other. Whereas gateway was placed at the height of 1.5m. on the opposite wall facing towards the echobeacons. Zone area of 1.5 x1.75 m^2 was separated with free-standing Screens and a chair was placed in each of the zones. Two power sources of 12 V DC were arranged, one for the Gateway and another for two echobeacons. The echobeacons were connected to the source in parallel. Four beacons were used as targets to be localized.

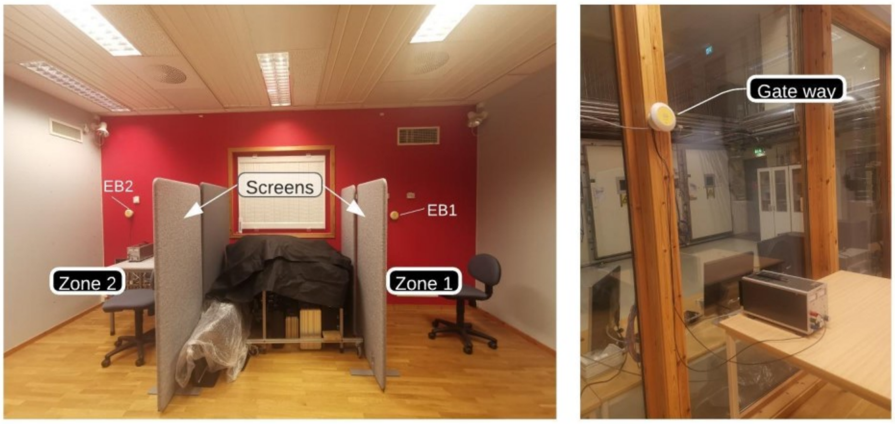


Figure 21 Experimental setup for indoor localization

The experiment was conducted for two separate localization types, assets, and personnel. Therefore 2 of the beacons were used for each set up. For assets localization one of the beacons was attached to a backpack and another was placed in a bucket. Similarly, for personnel localization the beacons were attached to wearables (wristband and keyrings). Figure 22 shows image of the targets. The targets where placed on the chair in the zones for collection of their localization data. The whole experiment setups were recorded with real time clock in the video, in order to tally physical presence of the target with the visualized presence in virtual platform.

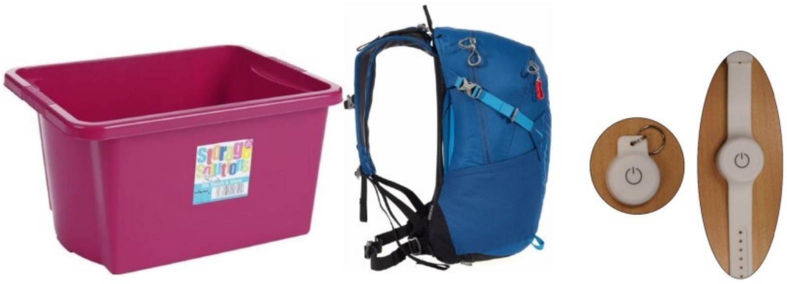
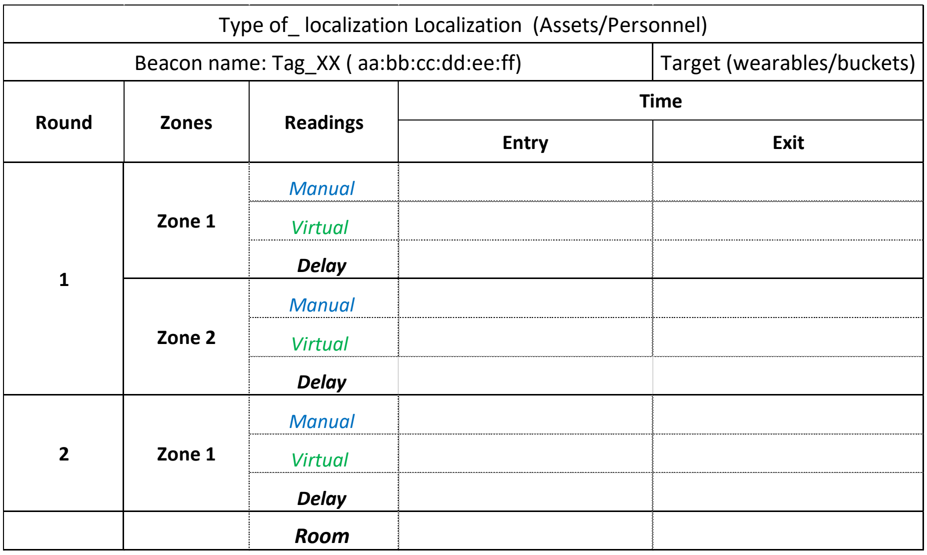


Figure 22 Targets for Assets and Personnel Localization

2.4.2.1 Procedure for collection of data:

a. Based upon data to be collected an observation table was created as shown in Figure 23. It contained two reading fields, Manual and Virtual to be filled with entry and exit time to each zone.



Type of_ localization Localization (Assets/Personnel)				
Beacon name: Tag_XX (aa:bb:cc:dd:ee:ff)				Target (wearables/buckets)
Round	**Zones**	**Readings**	**Time**	
			Entry	**Exit**
1	**Zone 1**	*Manual*		
		Virtual		
		Delay		
	Zone 2	*Manual*		
		Virtual		
		Delay		
2	**Zone 1**	*Manual*		
		Virtual		
		Delay		
		Room		

Figure 23 Observation table for target localization

b. For Manual readings the entry and exit point of the zones were defined by length of free-standing screens. Such that the targets were considered to have entered the zone if it was inside the area bounded by the screens and opposite walls. The corresponding time readings were taken and recorded in the table manually.

c. For virtual readings, the entry time was defined as point of time when the zone was highlighted in the test room model, and the target's tag name was shown on the respective zone in the visualization panel of the model. Similarly, the exit time was defined as the point of time when the target was last seen in the zone before it has been moved and visualized in the next zone. These data were taken from the database query and recorded video of the experiment.

d. The time difference between the manual readings and virtual readings were then used to calculate delay for each reading.

2.4.2.2 Target deployment Strategy:

The targets were deployed to collect two sets of data for each type of localization. In first set, only one target was considered, and observations were recorded. Similarly, on the second set both the targets were used, and recordings were made.

The strategy of target deployment for single target in both the localization types were as follow:

a. Turn ON the beacon attached to target & enter the room,
b. Enter to Zone1, wait there for a while.
c. Move to Zone2 , wait there for a while.
d. Come back to Zone1, wait there for a while.
e. Exit the room and turn OFF the beacon.

The strategy of target deployment for double targets in assets localization:

a. Turn ON the beacons attached to targets & Enter the Room,
b. Place both the targets in Zone 1, wait there for a while.
c. Remove one of the targets from Zone1 and Place it in Zone 2. Wait a while.
d. Move the remaining Target from Zone 1 to Zone 2, wait a while.
e. Move both the targets from Zone 2 to Zone 1, wait a while.
f. Exit the Room and turn OFF the beacons.

Similarly, strategy of target deployment for two people with wearables in personnel localization

a. Person 1 turns ON the beacon attached to his wearable & enters the Room,
b. Goes to Zone 1 and wait there for a while,

c. After some time, Person 2 turns ON the beacon attached to his wearable & enters the Room, goes to Zone 2 and wait there for a while.
d. Person 1 moves to Zone 2 and wait there for a while.
e. Both of them move to Zone 1 and wait there for a while.
f. Both of them exit the zone and turn OFF the beacons.

2.4.2.3 Observations and Data analysis:

Main purpose of the experiment was to test accuracy of the positioning device for the developed system, test the reliability of the generated data from the device and estimate overall delay of the system in the given set up. These were defined as follows.

Accuracy: The accuracy of the device for the system was percentage of measurements within which the devices were able to detect the target. It was calculated using following relation.

$$Accuracy = \frac{Available\ readings}{Total\ number\ of\ readings}\ X\ 100\% \qquad (6)$$

Reliability: Reliability referred to percentage measurement within which the system effectively detected the target in the locations and visualized them. Following relation was used for its calculation.

$$Reliability = \frac{Available\ readings-(False\ readings+overlap\ readings)}{Total\ number\ of\ readings}\ X\ 100\% \qquad (7)$$

Therefore, it was necessary to find the parameters constituting the relation shown is 6 & 7. For this collected data from observation tables of each target were thoroughly analysed using the recorded video of the experiment and database query. These parameters were defined as follows.

a. Available reading: These are the number of readings that were visualized by the system in the model. The count of available readings was generated from the database using SQL query.
b. Number of Blind readings: The readings were counted as Blind readings if the beacons were present in the zone but were not visualized by the system in the model. These readings were counted manually when message of 'no recent data in the zone' was displayed in visualization panel even if the targets were present.
c. Total number of readings: It is the sum of available readings and blind readings.
d. Number of False readings: The readings were counted as false readings if the targets were visualized in the zone by the system but were not present in the zone. These readings were

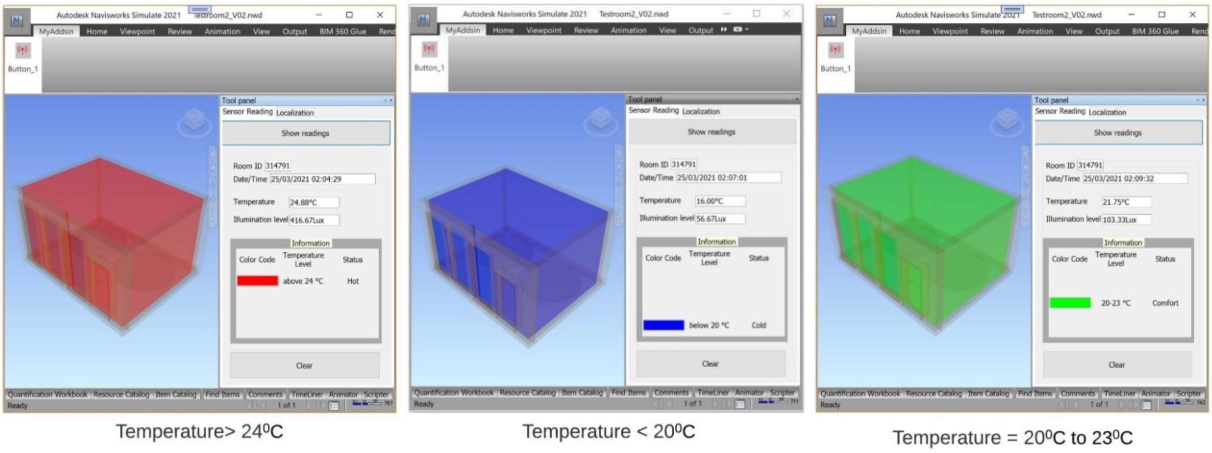
counted manually when the plugin displayed the tag name of the target and highlighted the zones even if it was not there.

e. Overlap readings: These are the counts of readings for which a target is visualized in both of the zones at the same time. These readings were counted manually.

3 Results and Discussions

3.1 Facility management data visualizations and analysis:

When temperature variations were made as described in section 2.4.1 following visualization results were observed.



Room ID 314791
Date/Time 25/03/2021 02:04:29
Temperature 24.88°C
Illumination level 416.67Lux

Information

Color Code	Temperature Level	Status
🟥	above 24 °C	Hot

Temperature> 24^{0}C

Room ID 314791
Date/Time 25/03/2021 02:07:01
Temperature 16.00°C
Illumination level 56.67Lux

Information

Color Code	Temperature Level	Status
🟦	below 20 °C	Cold

Temperature < 20^{0}C

Room ID 314791
Date/Time 25/03/2021 02:09:32
Temperature 21.75°C
Illumination level 103.33Lux

Information

Color Code	Temperature Level	Status
🟩	20-23 °C	Comfort

Temperature = 20^{0}C to 23^{0}C

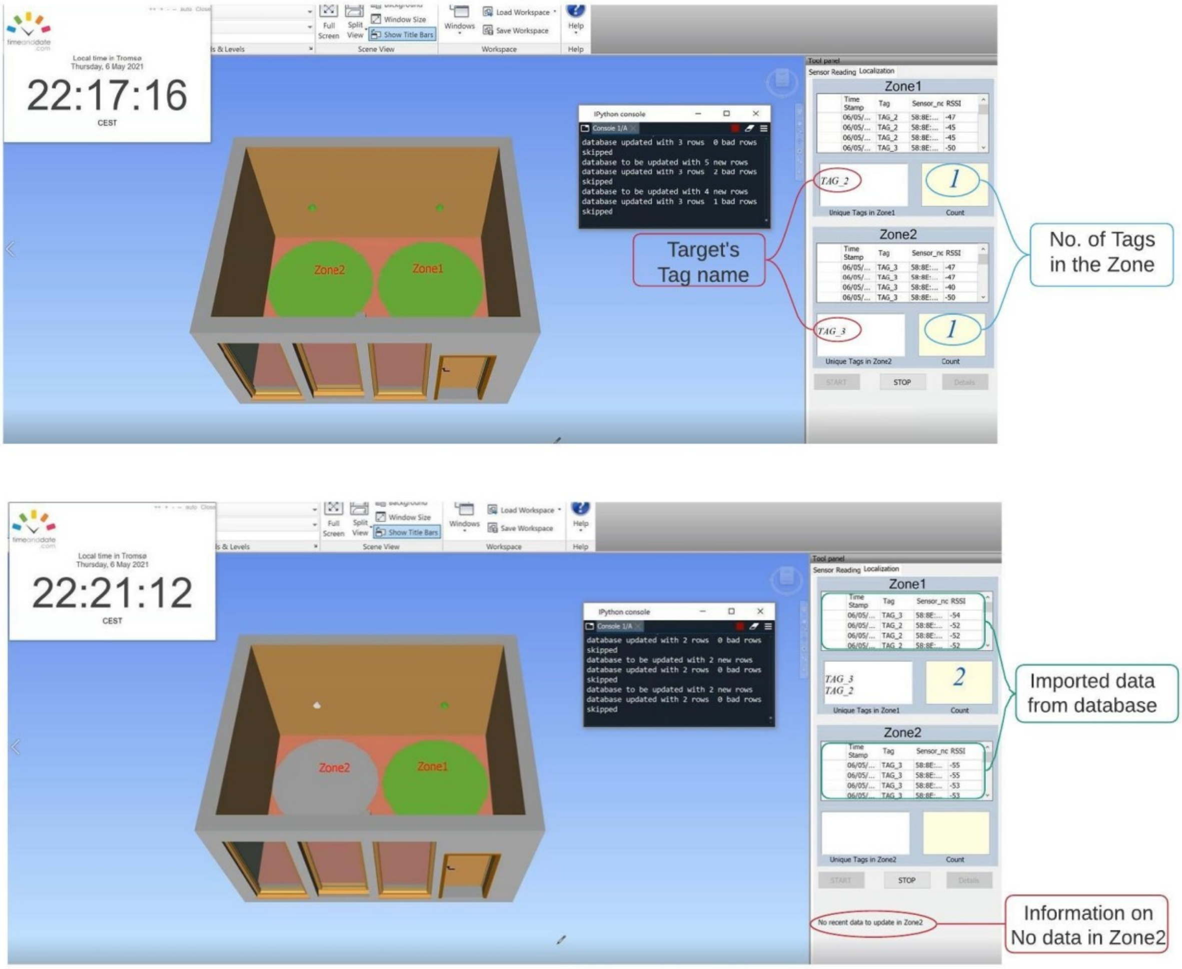
Figure 24 Visualization Results for Facility Management

Figure 24 shows visualization of Test room model in Navisworks platform when temperature variations were made. It was seen that whenever the temperature value was kept above 24^{0}C the test room model color changed to red, displaying 'Hot' status in the information table. Similarly, when temperature value was kept below 20^{0}C the test room color changed to blue displaying 'Cold' status in the information table and lastly, when temperature value was between 20^{0}C to 24^{0}C, the room color changed to green displaying 'Comfort' status for the room.

The displayed color-coded results were observed to be satisfying the desired output as per the programs written for facility management parts.

3.2 Localization data visualizations:

When targets were deployed to the respective zones, the developed system visualized the presence of the target by highlighting the zones with green color in the model. It also displayed the details showing the target's Tag name, counts and imported data. Figure 25 below shows the visualized data.



IPython console

```
database updated with 3 rows  0 bad rows
skipped
database to be updated with 5 new rows
database updated with 3 rows  2 bad rows
skipped
database to be updated with 4 new rows
database updated with 3 rows  1 bad rows
skipped
```

Time Stamp	Tag	Sensor_nc	RSSI
06/05/...	TAG_2	58:8E:...	-47
06/05/...	TAG_2	58:8E:...	-45
06/05/...	TAG_2	58:8E:...	-45
06/05/...	TAG_3	58:8E:...	-50

Unique Tags in Zone1	Count
TAG_2	1

Time Stamp	Tag	Sensor_nc	RSSI
06/05/...	TAG_3	58:8E:...	-47
06/05/...	TAG_3	58:8E:...	-47
06/05/...	TAG_3	58:8E:...	-40
06/05/...	TAG_3	58:8E:...	-50

Unique Tags in Zone2	Count
TAG_3	1

IPython console

```
database updated with 2 rows  0 bad rows
skipped
database to be updated with 2 new rows
database updated with 2 rows  0 bad rows
skipped
database to be updated with 2 new rows
database updated with 2 rows  0 bad rows
skipped
```

Time Stamp	Tag	Sensor_nc	RSSI
06/05/...	TAG_3	58:8E:...	-54
06/05/...	TAG_2	58:8E:...	-52
06/05/...	TAG_2	58:8E:...	-52
06/05/...	TAG_2	58:8E:...	-52

Unique Tags in Zone1	Count
TAG_3 TAG_2	2

Time Stamp	Tag	Sensor_nc	RSSI
06/05/...	TAG_3	58:8E:...	-55
06/05/...	TAG_3	58:8E:...	-55
06/05/...	TAG_3	58:8E:...	-53
06/05/...	TAG_3	58:8E:...	-53

No recent data to update in Zone2

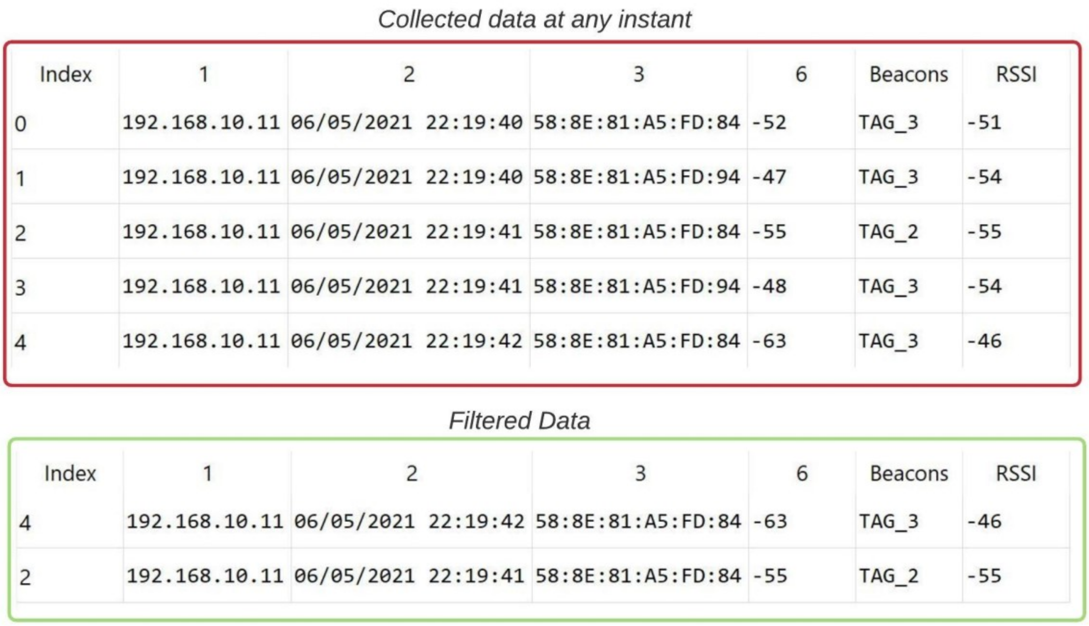
Figure 25 Visualization of Localization data

3.2.1 Result Analysis:

From the recorded data in observation tables and analysis of several parameters as discussed in section 2.4.2.3. performance of the system was measured. It was based on accuracy of the device, reliability of the system and variations in delay. Therefore, results obtained from single target from both types of localization (Assets and Personnel) were combined to find overall results. Similarly, results obtained from double target localization were combined and variation in the results were studied to determine the performance of the system for different number of targets. Following results were obtained.

a. Single target: Both over all Accuracy and Reliability of the system was found to be 87.96 %. The obtained values resulted to be same as no false and overlap readings were seen with single target localizations. Only blind readings were obtained with one-person localization. The blind reading could have been resulted when zone sensors were not able to receive the signals from the target. It is to be noted that the zone sensors were configured to receive signals from the target with RSSI value greater than -55db any target with RSSI values less than -55db are filtered out. The phenomena of multipath fading are often observed in small and confined room where the received signal strengths fluctuates rapidly, and the received signal level could be both constructive or destructive due to the characteristic of multipath fading signals arriving at the receiver antenna [63]. This could have resulted in lower RSSI value or signal strengths, or the placement of person could have been little farther away from the sensor range in the first set up. The delay or the response time for the system varied from 1 to 5 sec with single target arrangement where 5 secs was rare occurrence.

b. Double target: The overall accuracy of the system was found to be 93.11 % and Reliability was 89.89%. Both false and overlap readings were obtained in this set up. The false reading could have been resulted due to higher RSSI values or strong signal strength of the target received by the corresponding zone sensors. Usually, the sensor receives strong signal strength from the target nearest to it and data filter program filters the data based upon their RSSI values and allocate it to respective zones as shown Figure 26. However, at some instant if the signal strengths received by another zone sensor is higher than the signal strength of the sensors nearerst to the target false readings are obtained. Such phenomena are common where the space are constraints and sensors are placed close to each other. Only 5 overlap readings were noted throughout the test and all with 2 people localization set up. The possible reasons could be due to equal signal strength received by both the zone sensors resulting in to same RSSI values, Because of this the program could not distinguish between them and overlap readings are seen. The condition for overlap is illustrated in Figure 27. The response time of system with double target was found to be varying from 1 to 22 sec, this could be due to congestion of the data in gateway buffer, with more readings. This can be minimized through proper configuration of the device parameters like scan an advertising interval.



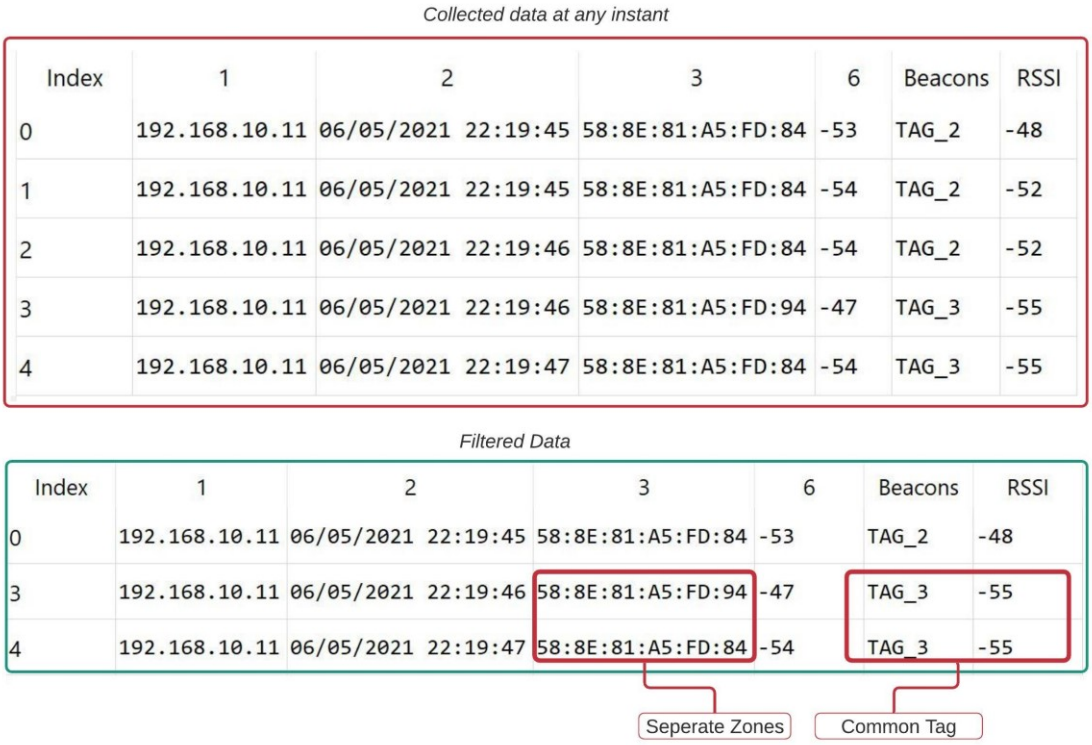
Collected data at any instant

Index	1	2	3	6	Beacons	RSSI
0	192.168.10.11	06/05/2021 22:19:40	58:8E:81:A5:FD:84	-52	TAG_3	-51
1	192.168.10.11	06/05/2021 22:19:40	58:8E:81:A5:FD:94	-47	TAG_3	-54
2	192.168.10.11	06/05/2021 22:19:41	58:8E:81:A5:FD:84	-55	TAG_2	-55
3	192.168.10.11	06/05/2021 22:19:41	58:8E:81:A5:FD:94	-48	TAG_3	-54
4	192.168.10.11	06/05/2021 22:19:42	58:8E:81:A5:FD:84	-63	TAG_3	-46

Filtered Data

Index	1	2	3	6	Beacons	RSSI
4	192.168.10.11	06/05/2021 22:19:42	58:8E:81:A5:FD:84	-63	TAG_3	-46
2	192.168.10.11	06/05/2021 22:19:41	58:8E:81:A5:FD:84	-55	TAG_2	-55

Figure 26 Application of RSSI filter



Collected data at any instant

Index	1	2	3	6	Beacons	RSSI
0	192.168.10.11	06/05/2021 22:19:45	58:8E:81:A5:FD:84	-53	TAG_2	-48
1	192.168.10.11	06/05/2021 22:19:45	58:8E:81:A5:FD:84	-54	TAG_2	-52
2	192.168.10.11	06/05/2021 22:19:46	58:8E:81:A5:FD:84	-54	TAG_2	-52
3	192.168.10.11	06/05/2021 22:19:46	58:8E:81:A5:FD:94	-47	TAG_3	-55
4	192.168.10.11	06/05/2021 22:19:47	58:8E:81:A5:FD:84	-54	TAG_3	-55

Filtered Data

Index	1	2	3	6	Beacons	RSSI
0	192.168.10.11	06/05/2021 22:19:45	58:8E:81:A5:FD:84	-53	TAG_2	-48
3	192.168.10.11	06/05/2021 22:19:46	58:8E:81:A5:FD:94	-47	TAG_3	-55
4	192.168.10.11	06/05/2021 22:19:47	58:8E:81:A5:FD:84	-54	TAG_3	-55

Figure 27 Condition for Overlap

From observation of the results, it can be seen that there is a little variation in reliability of the system for both single and double target localization. Which makes it a reliable system to effectively detect the location of the target with approximately 90% truth rate. The Accuracy of the system to detect the target was found to be comparatively higher with double target localization. This was due to higher numbers of blind readings in one person localization test.

The beacons attached to wearables had been configured with higher transmission power compared to beacons attached to assets, however higher blind readings were observed due to multipath fading. Also, it was noted that with proper configuration of the devices and sufficient space arrangements the performance of the proximity based indoor localization system can be increased.

The observation table recorded for each of the target and calculations are attached to Appendix E of the report.

3.2.2 Discussions:

The Integration framework developed during the case study consisted of several tools and platforms that supported data exchange and enabled access of functionalities from one application to another. The framework developed validates and comprehend conceptual frameworks discussed in [20]. It has modeled different IoT Gateway, using Raspberry PI and BluEpyc device. The gateway intermediates the transfer of data from one layer to another layer. Facility management part of the project utilized connection string of virtual IoT device registered in Azure IoT Hub to transfer data using Raspberry pi module as a Gateway and TCP/IP communication protocol intermediated by python program. Whereas in the second case the indoor positioning device's gateway was used with TCP/IP protocol to transfer the localization data to csv file in local PC. This shows variations on gateway functionalities based upon tools and technologies being used. The csv file was transferred to local SQL server using a separate python program in the PC. This program utilized 'pyodbc' libraries to connect with the database using its connection string.

The data visualization technique and middleware used in the study were based on .NET Framework 4.7 that used C# & Visual studio as programing platform. ADO.NET is a set of class that provides consistent access to data sources like SQL Servers and includes .NET Framework data providers for connecting to database, executing commands, and retrieving results [64]. These functionalities have been used to retrieve data from the database server and used it together with plugins developed for Navisworks for visualization of both environment and localization data in a common platform. However, the developed integrated system requires, simultaneous running of different programs to stream data in real time. Using Azure SQL server as cloud database requires running of stream analytics to connect physical IoT device through IoT Hub to Azure SQL server and should be run continuously as long as real time data are desired to be obtained. Therefore, to obtain economic usability of the service the data can be passed to cloud system only when readings changes by some limits and similar functionalities can be added to IoT devices. The panda library in python program used in localization part of the case study to filter data has proven to be an efficient tool for data analysis and filtering of unwanted data. It seamlessly allocated the data to respective sensors zones based on higher RSSI values received from the beacons. The localization device based on Bluetooth technology implemented in the case study needs to be configured prior to implementations and are often dependent to application strategy, range selections and desired scan interval. These parameters can be varied for each component (beacon, echobeacons & gateway). This may often result in congestion of data in the gateway when the data arrival rate is more than its processing rate. In the study similar delays were seen for some cases with double target localization. Navisworks API for development of plugins to visualize IoT data has been found to be an effective tool that enable visualization of multiple data from different IoT devices in real time and also access the historical data. However, the study has not fully tapped into other potential area of localization such as tracking the assets based on their coordinates and distance computations which could be more intuitive approach. Instead, the study has presented the proximity-based visualizations to provide information on people presence in the allocated region which is also the common practice of localization using BIM and IoT tools [23, 25]. There were also some limitations regarding spaces. It was found to be smaller for sensors set ups. High transmission signals from the beacons in a smaller space are subjected to multipath fading and rapid fluctuations on signal strengths this could have influenced error on the measurements. Also, manual readings in the observation's tables are subjected to minor deviations due to real time measurements and human error.

Some of the challenges related to the study and its implementation are use of Navisworks simulate software for facility management and localization process. As these softwares are primarily being used in construction industries to combine 3D models, detect clash, review, and simulate construction process. Using it for data visualization is new aspect of utilizations. Also, to localize personnel or assets using Bluetooth technology, requires deployment of beacons, or registering of Bluetooth device to the system, . therefore, localization is only limited to targeted group people or assets equipped with beacons. Using such devices with other applications also requires an integrating program to be developed based on the application to be used.

4 Conclusions and Future works

BIM tools and IoT devices used in the study have provided an overview on how these technologies communicate with each other for data exchange and information access. The study has highlighted four basic blocks required for development of an integrated system using BIM and IoT tools. The project presented these blocks as IoT block for data generation, filter and transfer, database server block for data storage & management, middleware block for data import & integration with softwares, and application block for data visualizations through the software. From the study of two different IoT devices together with BIM based system, it can be said that IoT devices, often require, sensors working together with single board computer (Raspberry Pi) or microcontroller devices for data filtration and data transfer to database. The connection to these databases can be made by importing libraries based on data access APIs to the programing interface of single board devices. This enables connection to database with connection strings. Both cloud based and local data server could be utilized for data storage and management. Tools and designers in Visual studio working together with ADO.NET could be used for connection to database and retrieval of data. It was found that the APIs of the visualizing software were the links utilized by the developed platforms to integrate imported data with the BIM software.

The project developed a common platform for visualization of different IoT generated data. It also implemented several data filtering techniques. It was also able to provide access to historical localization data based on entry and exit time of the targets. There has not been found much work on indoor localization using Navisworks platform. Very few of the works are available where the Navisworks has been modeled for the purpose of facility management. This thesis has utilized the idea of facility management to visualize the localization of assets and personnel in an indoor area. The result from the observations showed that the system has higher reliability and are suitable for implementation in larger scale provided adequate separation between receiving sensors with proximity-based localization method.

The BIM & IoT integration have higher potential to digitalize different physical process based upon their utilization domains. In context of facility management, it can be used to control the field devices as well. This thesis has only presented data visualization using Illumination and temperature sensors. Similar, techniques to measure ambient light entering the room and using them to balance the total indoor light by varying the light intensity from the light fixtures can help in energy savings. Such variation can be made using actuator devices linked to dimmer switch of the light. This can also be applied with heating and ventilation equipment making the buildings Smart enough to adopt with outer environments. Also, the cloud-based BIM services are new concept and are gradually developing [2]. Using such technology development of browser-based monitoring platform could be more beneficial for remote monitoring. Such aspects of implementations can be the future research for this project.

5 References

[1] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *2012 10th international conference on frontiers of information technology*, 2012: IEEE, pp. 257-260.

[2] U. Isikdag, "BIM and IoT: A synopsis from GIS perspective," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences,* vol. 40, p. 33, 2015.

[3] K. Ozsoy, A. Bozkurt, and I. Tekin, "Indoor positioning based on global positioning system signals," *Microwave and Optical Technology Letters,* vol. 55, no. 5, pp. 1091-1097, 2013.

[4] S. Ramnath, A. Javali, B. Narang, P. Mishra, and S. K. Routray, "IoT based localization and tracking," in *2017 International Conference on IoT and Application (ICIOT)*, 2017: IEEE, pp. 1-4.

[5] J. L. Hernández, P. Martín Lerones, P. Bonsma, A. Van Delft, R. Deighton, and J.-D. Braun, "An IFC interoperability framework for self-inspection process in buildings," *Buildings,* vol. 8, no. 2, p. 32, 2018.

[6] C. M. Eastman, C. Eastman, P. Teicholz, R. Sacks, and K. Liston, *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2011.

[7] J. Wong, X. Wang, H. Li, and G. Chan, "A review of cloud-based BIM technology in the construction sector," *Journal of information technology in construction,* vol. 19, pp. 281-291, 2014.

[8] J. Underwood and U. Isikdag, "Emerging technologies for BIM 2.0," *Construction Innovation,* 2011.

[9] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems,* vol. 29, no. 7, pp. 1645-1660, 2013.

[10] I. Butun, *Industrial IoT*. Springer, 2020.

[11] K. Schwab, *The fourth industrial revolution*. Currency, 2017.

[12] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer Networks,* vol. 144, pp. 17-39, 2018.

[13] J.-P. Calbimonte, S. Sarni, J. Eberle, and K. Aberer, "XGSN: An Open-source Semantic Sensing Middleware for the Web of Things," in *TC/SSN@ ISWC*, 2014, pp. 51-66.

[14] J. Guth, U. Breitenbücher, M. Falkenthal, F. Leymann, and L. Reinfurt, "Comparison of IoT platform architectures: A field study based on a reference architecture," in *2016 Cloudification of the Internet of Things (CIoT)*, 2016: IEEE, pp. 1-6.

[15] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th-18th January, 2014*, 2014: IEEE, pp. 414-419.

[16] L. Wang *et al.*, "Cloud computing: a perspective study," *New generation computing,* vol. 28, no. 2, pp. 137-146, 2010.

[17] D. Sasikumar and S. Saravanakumar, "Cloud computing–Research issues, challenges, architecture, platforms and applications," *International Research Journal of Latest Trends in Engineering and Technology,* vol. 5, no. 1, pp. 39-46, 2017.

[18] K. Ferguson-Boucher, "Cloud computing: A records and information management perspective," *IEEE Security & Privacy,* vol. 9, no. 6, pp. 63-66, 2011.

[19] W. Kim, "Cloud computing: Today and tomorrow," *J. Object Technol.,* vol. 8, no. 1, pp. 65-72, 2009.

[20] M. Shahinmoghadam and A. Motamedi, "Review of BIM-centred IoT Deployment–State of the Art, Opportunities, and Challenges," in *Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC 2019)*, 2019, pp. 1268-1275.

[21] B. Dave, A. Buda, A. Nurminen, and K. Främling, "A framework for integrating BIM and IoT through open standards," *Automation in Construction,* vol. 95, pp. 35-45, 2018.

[22] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends," *Automation in Construction,* vol. 101, pp. 127-139, 2019.

[23] J. Teizer *et al.*, "Internet of Things (IoT) for integrating environmental and localization data in Building Information Modeling (BIM)," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 2017, vol. 34: IAARC Publications.

[24] M. Neges, M. Wolf, M. Propach, J. Teizer, and M. Abramovici, "Improving indoor location tracking quality for construction and facility management," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 2017, vol. 34: IAARC Publications.

[25] Y. Fang, Y. K. Cho, S. Zhang, and E. Perez, "Case study of BIM and cloud–enabled real-time RFID indoor localization for construction management applications," *Journal of Construction Engineering and Management,* vol. 142, no. 7, p. 05016003, 2016.

[26] M.-Y. Cheng, K.-C. Chiu, Y.-M. Hsieh, I.-T. Yang, J.-S. Chou, and Y.-W. Wu, "BIM integrated smart monitoring technique for building fire prevention and disaster relief," *Automation in Construction,* vol. 84, pp. 14-30, 2017.

[27] K. M. Rashid, J. Louis, and K. K. Fiawoyife, "Wireless electric appliance control for smart buildings using indoor location tracking and BIM-based virtual environments," *Automation in Construction,* vol. 101, pp. 48-58, 2019.

[28] B. Mataloto, J. C. Ferreira, R. Resende, R. Moura, and S. Luís, "BIM in People2People and Things2People Interactive Process," *Sensors,* vol. 20, no. 10, p. 2982, 2020.

[29] K.-M. Chang, R.-J. Dzeng, and Y.-J. Wu, "An automated IoT visualization BIM platform for decision support in facilities management," *Applied sciences,* vol. 8, no. 7, p. 1086, 2018.

[30] D. Kazado, M. Kavgic, and R. Eskicioglu, "Integrating Building Information Modeling (BIM) and sensor technology for Facility Management," *Journal of Information Technology in Construction (ITcon),* vol. 24, no. 23, pp. 440-458, 2019.

[31] A. Vatankhah Barenji and M. Hashemipour, "Real-time building information modeling (BIM) synchronization using radio frequency identification technology and cloud computing system," *Journal of Industrial and Systems Engineering,* vol. 10, no. 1, pp. 61-68, 2017.

[32] C. S. Araújo, L. C. de Siqueira, E. d. A. M. Ferreira, and D. B. Costa, "Conceptual Framework for Tracking Metallic Formworks on Construction Sites Using IoT, RFID and BIM Technologies," in *International Conference on Computing in Civil and Building Engineering*, 2020: Springer, pp. 865-878.

[33] F. Xue, K. Chen, W. Lu, Y. Niu, and G. Huang, "Linking radio-frequency identification to Building Information Modeling: Status quo, development trajectory and guidelines for practitioners," *Automation in Construction,* vol. 93, pp. 241-251, 2018.

[34] D. Macagnano, G. Destino, and G. Abreu, "Indoor positioning: A key enabling technology for IoT applications," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014: IEEE, pp. 117-118.

[35] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *computer,* vol. 34, no. 8, pp. 57-66, 2001.

[36] C.-C. Pu, C.-H. Pu, and H.-J. Lee, "Indoor location tracking using received signal strength indicator," *Emerging communications for wireless sensor networks,* p. 11, 2011.

[37] M. Abdalla, S. M. Feeney, and S. Salous, "Antenna array and quadrature calibration for angle of arrival estimation," *SCI2003, Florida,* 2003.

[38] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking moving devices with the cricket location system," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, 2004, pp. 190-202.

[39] K. Pahlavan, P. Krishnamurthy, and A. Beneat, "Wideband radio propagation modeling for indoor geolocation applications," *IEEE Communications Magazine,* vol. 36, no. 4, pp. 60-65, 1998.

[40] K. Pahlavan, X. Li, and J.-P. Makela, "Indoor geolocation science and technology," *IEEE communications magazine,* vol. 40, no. 2, pp. 112-118, 2002.

[41] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 37, no. 6, pp. 1067-1080, 2007.

[42] R.-H. Lee, J.-C. Wu, S.-H. Chang, S.-F. Chang, C.-C. Chang, and Y.-M. Chen, "Radar design for wireless indoor positioning applications," in *2013 European Microwave Conference*, 2013: IEEE, pp. 846-849.

[43] G. Shi and Y. Ming, "Survey of indoor positioning systems based on ultra-wideband (UWB) technology," in *Wireless Communications, Networking and Applications*: Springer, 2016, pp. 1269-1278.

[44] B. O'Keefe, "Finding Location with Time of Arrival and Time Difference of Arrival Techniques," *ECE Senior Capstone Project,* 2017.

[45] G. Shen, R. Zetik, and R. S. Thoma, "Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment," in *2008 5th Workshop on Positioning, Navigation and Communication*, 2008: IEEE, pp. 71-78.

[46] H.-S. Ahn and W. Yu, "Environmental-adaptive RSSI-based indoor localization," *IEEE Transactions on Automation Science and Engineering,* vol. 6, no. 4, pp. 626-633, 2009.

[47] D. Stojanović and N. Stojanović, "Indoor localization and tracking: Methods, technologies and research challenges," *Facta Universitatis, Series: Automatic Control and Robotics,* vol. 13, no. 1, pp. 57-72, 2014.

[48] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. Fangmeyer, "Evolution of indoor positioning technologies: A survey," *Journal of Sensors,* vol. 2017, 2017.

[49] "RFID Basics." https://www.brooks.com/applications-by-industry/semiconductor/rfid/rfid-basics (accessed 2021).

[50] K. Ahsan, H. Shah, and P. Kingston, "RFID applications: An introductory and exploratory study," *arXiv preprint arXiv:1002.1179,* 2010.

[51] G. De Blasio, A. Quesada-Arencibia, C. R. García, J. C. Rodriguez-Rodriguez, and R. Moreno-Díaz, "A protocol-channel-based indoor positioning performance study for Bluetooth low energy," *IEEE Access,* vol. 6, pp. 33440-33450, 2018.

[52] J. Powar, C. Gao, and R. Harle, "Assessing the impact of multi-channel BLE beacons on fingerprint-based positioning," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2017: IEEE, pp. 1-8.

[53] V. Cantón Paterna, A. Calveras Auge, J. Paradells Aspas, and M. A. Perez Bullones, "A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering," *Sensors,* vol. 17, no. 12, p. 2927, 2017.

[54] K. Cisek, A. Zolich, K. Klausen, and T. A. Johansen, "Ultra-wide band Real time Location Systems: Practical implementation and UAV performance evaluation," in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, 2017: IEEE, pp. 204-209.

[55] S. Ullah, M. Ali, A. Hussain, and K. S. Kwak, "Applications of UWB technology," *arXiv preprint arXiv:0911.1681,* 2009.

[56] A. Alarifi *et al.*, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors,* vol. 16, no. 5, p. 707, 2016.

[57] T. A. Alhmiedat and S.-H. Yang, "A ZigBee-based mobile tracking system through wireless sensor networks," *International Journal of Advanced Mechatronic Systems,* vol. 1, no. 1, pp. 63-70, 2008.

[58] S. Tekinay, E. Chao, and R. Richton, "Performance benchmarking for wireless location systems," *IEEE Communications Magazine,* vol. 36, no. 4, pp. 72-76, 1998.

[59] J. Liu, "Survey of wireless based indoor localization technologies," *Department of Science & Engineering, Washington University,* 2014.

[60] S. Campbell. "Basics of the I2C Communication Protocol." https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/#comments (accessed 2021).

[61] R. Stackowiak, "Azure IoT Hub," in *Azure Internet of Things Revealed: Architecture and Fundamentals*. Berkeley, CA: Apress, 2019, pp. 73-85.

[62] S. Klein, "Azure Stream Analytics," in *IoT Solutions in Microsoft's Azure IoT Suite: Data Acquisition and Analysis in the Real World*. Berkeley, CA: Apress, 2017, pp. 71-84.

[63] T. S. Rappaport, "Characterization of UHF multipath radio channels in factory buildings," *IEEE transactions on antennas and propagation,* vol. 37, no. 8, pp. 1058-1069, 1989.

[64] M. Doumentation. "ADO.NET Overview." https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview (accessed 11/05/2021.

Appendix A

Python program in Raspberry Pi module for transfer of Sensor data to Azure Iot hub.

1. Raspberry Pi configurations

   sudo apt update

   sudo apt install python3 idle 3 (Install python idle 3)

   sudo raspi-config (GUI to configure and enable I2c port using)

2. installing required library

   sudo-apt-get install -y

   python-smbus i2c-tools

   pip3 install azure-iot-hub-device-client

   pip install asyncio

3. Creating I2c bus number 2

   cd /boot

   sudo nano config.txt

   dtoverlay =i2c-gpio,bus=2,i2c_gpio_sda=22,i2c_gpio_scl=23 (written below the line where i2c & smp are enabled)

   ctrl +X & Y (to save)

   cd ~¨

   Reboot the device,

4. Connection diagram of Sensors to Raspberry Pi terminals

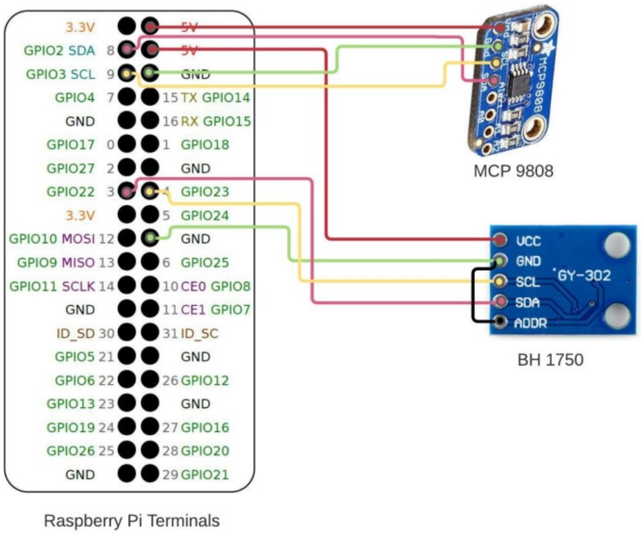


Figure i: Connection diagram of Raspberry pi terminals to Sensors

5. Python programming part

```
import smbus
import time
import asyncio
import json
import datetime
from azure.iot.device.aio import IoTHubDeviceClient
#Get I2c bus
bus=smbus.SMBus(2)
bus2=smbus.SMBus(1)
#MCP9808 address, 0x18(24)
#select configuration register, 0x01(1)
# 0x0000(00) continuous conversion mode, power-up default
config=[0x00, 0x00]
bus2.write_i2c_block_data(0x18,0x01, config)
#MCP9808 address 0x18(24)
#select resolution register, 0x08(8)
# 0x03(03) Resolution = +0.0625 / C
bus2.write_byte_data(0x18,0x08,0x03)
async def main():
    conn_str="HostName=Raspberrypi4.azure-
devices.net;DeviceId=1111;SharedAccessKey=e/z2FgKNgTz6wBbTFv0oMsaJU4s9
GzBlnEga1qTva/s="
    device_client =IoTHubDeviceClient.create_from_connection_string(conn_str)
    await device_client.connect()

    while True:
        time.sleep(30)
#MCP9808 address, 0x18(24)
#Read data back from 0x05(5), 2 bytes
#Temp MSB, TEMP LSB
        data=bus2.read_i2c_block_data(0x18, 0x05, 2)
#Convert the data to 13 bits
        ctemp=((data[0]& 0x1F)* 256)+ data[1]
        if ctemp>4095:
            ctemp -=8192
        ctemp=ctemp *0.0625
        ftemp=ctemp*1.8+32
#BH1750 address, 0x23,0x23 is one time 1-res mode 0010_0011
        data=bus.read_i2c_block_data(0x23, 0x23)
#Convert the data to illuminance
        luminance =(data[0] * 256+data[1]) /1.2
#output data to screen
        print("Ambient light level: %2f lux" %luminance)
        print("\n Temperature in Celsius is :%2f C" %ctemp)
        print("\n Temperature in Fahrenheit is: %2f F" %ftemp)
        ElementId=314791
        data=[]
        data= {'timestamp':str(datetime.datetime.now()),
              'Temperature' : ctemp,
              'Lux' : luminance, 'RoomId' : ElementId
              }
        s=json.dumps(data)
        await device_client.send_message(s)
    await device_client.disconnect()

if __name__=="__main__":
    asyncio.run(main())
```

Appendix B

1. Creating Ribbon with XAML content page

```
<?xml version="1.0" encoding="utf-8" ?>
<RibbonControl
    x:Uid="AddinRibbon" xmlns="clr-namespace:Autodesk.Windows;assembly=AdWindows"
    xmlns:wpf="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:adwi="clr-namespace:Autodesk.Internal.Windows;assembly=AdWindows"
    xmlns:system="clr-namespace:System;assembly=mscorlib"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-
namespace:Autodesk.Navisworks.Gui.Roamer.AIRLook;assembly=navisworks.gui.roamer">
```

```
    <RibbonTab Id="ID_CustomTab_1" KeyTip="T1">
        <RibbonPanel x:Uid="RibbonPane1_1">
            <RibbonPanelSource x:Uid="RibbonPanelSource_RibbonPanel_1" KeyTip="C1"
Title="Tool Panel">
                <local:NWRibbonButton x:Uid="Sensors" Id="ID_Button_1"
                            Size="Large"
                            KeyTip="B1"
                            ShowText="True"
                            Text="Tool Panel"/>

            </RibbonPanelSource>
        </RibbonPanel>
    </RibbonTab>

</RibbonControl>
```

2. ***Creation of Dock panel and Adding user control Tab for environmental and localization sensors:,***

   Required Library References: Autodesk.Navisworks.Api

   ,Navisworks.gui.roamer,System.Data, Adwindows

```
using System;
using System.Windows.Forms;
using AddinRibbon;
using AddinRibbon.Ctr;
using Autodesk.Navisworks.Api.Plugins;
namespace AddinRibbon
{
    [Plugin("AddRibbon","Jeewa", DisplayName ="AddinRibbon")] //defining Plugin
    [RibbonLayout("AddinRibbon.xaml")]
    [RibbonTab("ID_CustomTab_1", DisplayName ="MyAddsin")]
    [Command("ID_Button_1", LargeIcon = "1-32.png", ToolTip ="Open the interface for
sensor data visualization")]

public class CIAddin : CommandHandlerPlugin
    {
        public override int ExecuteCommand(string name, params string[] parameters)
        {
            switch (name)
            {
                case "ID_Button_1":

                       if (!Autodesk.Navisworks.Api.Application.IsAutomated)
                       {
                           var pluginRecord =
Autodesk.Navisworks.Api.Application.Plugins.FindPlugin("ClDockPanelUpdate.jeewa");
                           if (pluginRecord is DockPanePluginRecord &&
pluginRecord.IsEnabled)
                           {
                               var docPanel =
(DockPanePlugin)(pluginRecord.LoadedPlugin ?? pluginRecord.LoadPlugin());
                               docPanel.ActivatePane();
                           }

                       }
```

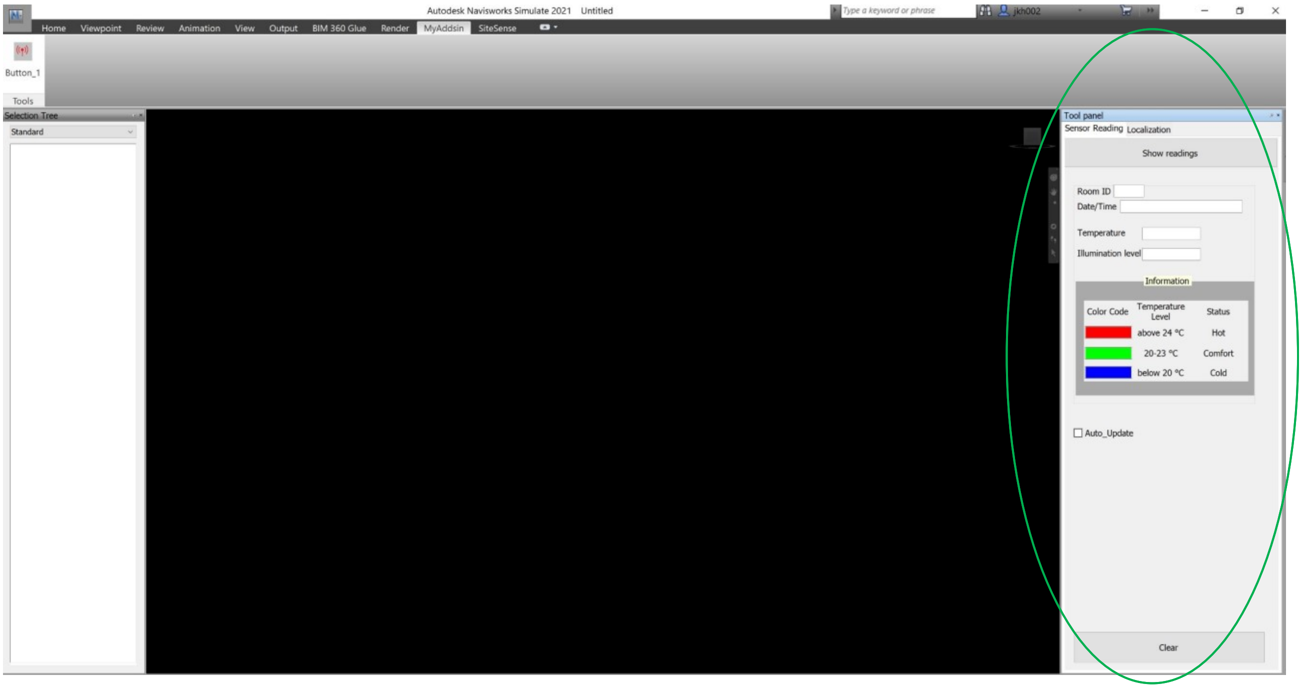
```
                    break;
            }
            return 0;
        }
    }
}
namespace AddinDockPanel
{
    [Plugin("ClDockPanelUpdate", "jeewa", DisplayName = "Tool panel")]
    [DockPanePlugin(200, 400, AutoScroll = true, MinimumHeight = 100, MinimumWidth =
200)]//Parameter setting width height...
    public class ClDockpanelUpdate : DockPanePlugin
    {
        public override Control CreateControlPane() //Override controlpane
        {
            var tc = new TabControl(); //creating new tab control in the control pane
            tc.ParentChanged += SetDockstyle;



            var tp3 = new TabPage("Sensor Reading");// Creating new tab for Sensors
Readings
            tp3.Controls.Add(new Sensors());
            tc.TabPages.Add(tp3);

            var tp4 = new TabPage("Localization");// Creating new tab for
Locallization Readings
            tp4.Controls.Add(new Locaization());
            tc.TabPages.Add(tp4);


            return tc;


        }

        private void SetDockstyle(object sender, EventArgs e)
        {
            try
            {
                var tc = sender as TabControl;
                tc.Dock = DockStyle.Fill; // Filling as parent changes
            }
            catch (Exception)
            {

            }
        }

        public override void DestroyControlPane(Control pane)//destroy controlpane
        {

            try
            {
                var ctr = (Sensors)pane;
                ctr?.Dispose();
            }
            catch (Exception)
            {
               //

            }
        }
    }
}
```

3. *Program for Temperature and illumination data visualizations sensors:*

```
using System;
using System.Windows.Forms;
using System.Data.SqlClient;
using Autodesk.Navisworks.Api;
using App = Autodesk.Navisworks.Api.Application;



namespace AddinRibbon.Ctr
{
    public partial class Sensors : UserControl
    {
        private ModelItemCollection Lastselection;

        protected override void OnParentChanged(EventArgs e)
        {
            base.OnParentChanged(e);
            Dock = DockStyle.Fill;
        }
        public Sensors()
        {
            InitializeComponent();

        }

        private void Readings_Click(object sender, EventArgs e)
        {
            Showreadings();
        }

        private void Showreadings()
        {
            try
            {
                if (App.ActiveDocument.FileName != "")
                {
                    Document oDoc = App.ActiveDocument;
                    ModelItem rootItem = oDoc.Models[0].RootItem;
                    if (rootItem.DisplayName == "Testroom2_V02.nwd")
                    {
                            string connstr =
                           "Server=tcp:readingserver.database.windows.net,1433;" +
                            "Initial Catalog=Readingdb;Persist Security Info=False;" +
                            "User ID=MCP9808;Password=Availablebh1750;" +
                            "MultipleActiveResultSets=False;Encrypt=True;" +
                            "TrustServerCertificate=False;Connection Timeout=30;";
                        using (var conn = new SqlConnection(connstr))
                        {
                            using (var command = conn.CreateCommand())
                            {
                                try
                                {
                                    command.CommandText = @"SELECT TOP (1) [RoomId],
                       [timestamp], CAST([Temperature] AS decimal(10,2)),
                                                    CAST([Lux] AS decimal(10,2)) FROM
                           readingtable1 order by Id Desc";
                                    conn.Open();
                                }
                                catch (Exception l)
                                {

                                     MessageBox.Show($"message {l}");
                                }

                                using (var reader = command.ExecuteReader())
                                {
                                    while (reader.Read())
                                    {
                                    var s = new Search();
                                    s.Locations = SearchLocations.DescendantsAndSelf;

                                    s.Selection.SelectAll();

            var sc = SearchCondition.HasPropertyByDisplayName("Element ID", "Value");

s.SearchConditions.Add(sc.EqualValue(VariantData.FromDisplayString(Convert.ToString(re
ader.GetValue(0))))); ;//Search condition to find Room element in model using element
ID value.
                                        var set1 = new ModelItemCollection();
                                        set1 = s.FindAll(oDoc, false);
                                        Lastselection = set1;
                                        oDoc.CurrentSelection.CopyFrom(Lastselection);
;// conditions to change color parameter of room.
                                        if ((Convert.ToDecimal(reader.GetValue(2)) >
20 && Convert.ToDecimal(reader.GetValue(2)) < 24))
                                        {

oDoc.Models.OverridePermanentTransparency(Lastselection, 0.5);

oDoc.Models.OverridePermanentColor(Lastselection, new Autodesk.Navisworks.Api.Color(0,
1, 0));
                                            oDoc.CurrentSelection.Clear();
                                            textBox1.Visible = false;
                                            label9.Visible = false;
                                            label12.Visible = false;

                                            textBox4.Visible = false
                                            label11.Visible = false;
                                            label14.Visible = false;
                                        }


else if (((Convert.ToDecimal(reader.GetValue(2)) < 20)))
                                        {

oDoc.Models.OverridePermanentTransparency(Lastselection, 0.5);

oDoc.Models.OverridePermanentColor(Lastselection, new Autodesk.Navisworks.Api.Color(0,
0, 1));
                                            oDoc.CurrentSelection.Clear();

                                            textBox1.Visible = false; //Red
                                            label9.Visible = false;
                                            label12.Visible = false;
                                            textBox3.Visible = false;//green
                                            label10.Visible = false;
                                            label13.Visible = false;
                                        }
                                        else
                                        {

oDoc.Models.OverridePermanentTransparency(Lastselection, 0.5);

oDoc.Models.OverridePermanentColor(Lastselection, new Autodesk.Navisworks.Api.Color(1, 
0, 0));
                                            oDoc.CurrentSelection.Clear();

                                            textBox3.Visible = false;//green
                                            label10.Visible = false;
                                            label13.Visible = false;

                                            textBox4.Visible = false; //Blue
                                            label11.Visible = false;
                                            label14.Visible = false;
                                        }

                                        oDoc.ActiveView.FocusOnCurrentSelection();
                                        oDoc.ActiveView.LookFromFrontRightTop();
                                        //SendKeys.Send("{HOME}");
                                        set1.Invert(oDoc);

oDoc.Models.OverrideTemporaryTransparency(set1, 0.8);
                                        try
                                        {

                                            rm_idbox.Text = 
string.Concat(reader.GetValue(0));
                                            date_box.Text = 
string.Concat(reader.GetValue(1));
                                            tempBox.Text = 
string.Concat(reader.GetValue(2), "°C");
                                            illum_box.Text = 
string.Concat(reader.GetValue(3), "Lux");
                                        }
                                        catch (Exception x)
                                        {

                                            MessageBox.Show($"Message! {x}");
                                        }


                                    }


                                 }
                               conn.Close();
                           }
                       }

                   }
                   else
                   {
                       MessageBox.Show("Open Testroom2_V02.nwd Model", "Message", 
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
                   }
```

```
                }
                else
                {
                    MessageBox.Show("No model detected", "Error", 
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            catch (Exception)
            {
                throw;
            }
        }

        private void Clearbtn_Click(object sender, EventArgs e)
        {
            rm_idbox.Clear();
            date_box.Clear();
            tempBox.Clear();
            illum_box.Clear();

            textBox1.Visible = true; //Red
            label9.Visible = true;
            label12.Visible = true;

            textBox3.Visible = true;//green
            label10.Visible = true;
            label13.Visible = true;

            textBox4.Visible = true; //Blue
            label11.Visible = true;
            label14.Visible = true;

            App.ActiveDocument.ActiveView.LookFromFrontRightTop();
            App.ActiveDocument.Models.ResetAllPermanentMaterials();
            App.ActiveDocument.Models.ResetAllTemporaryMaterials();
        }

    }
}
```

4. *Tool Panel Temperature and illumination data visualizations sensors:*



Color Code	Temperature Level	Status
🟥	above 24 °C	Hot
🟩	20-23 °C	Comfort
🟦	below 20 °C	Cold

Figure ii: Tool panel for visualization of temperature and illumination data

Appendix C

1. ***Python program for data filter and transfer from csv file to SQL server database.***

```
import time
import pandas as pd
import pyodbc as odbc

# c8:49:6d:cf:9a:b8 =TAG_1
# d4:32:46:39:8e:a3 =TAG_2
# d9:3c:4c:a1:ee:5a =TAG_3
# e8:a8:08:43:d0:25 =TAG_4
# f4:30:c7:04:a2:44 =TAG_5

pd.set_option('display.max_columns', 7)
pd.set_option("display.max_rows", None)
pd.set_option('display.width', 1000)
columns1 = ['TIME STAMP', 'MAC ADDRESS', 'EB_RSSI','Beacons','RSSI']
read_length =0


def datafilter(lf):
    dt=pd.read_csv(r'C:\Program Files
(x86)\BluEpyc\BluEpycStart\Log_06_05_2021.txt',

sep=';',header=None,warn_bad_lines=True,error_bad_lines=False)
    read_length = len(dt)
    new_data=read_length-10
    print("database to be updated with", new_data, "new rows")
    if  read_length > lf :

        tofilter=dt.tail(5).copy()

        rows_length=len(tofilter)
        x=[]#Empty list to store index of bad data
        tofilter.reset_index(drop=True, inplace=True)
        for i in range(rows_length):
         try:

tofilter[['Beacons','RSSI','bx','cx','xx']]=tofilter.loc[[i],4].str.split('
',expand=True,) #only data that resembles desired format
         except ValueError:
          x.append(i) # appending data index with bad formats
          continue
        valid_rows =tofilter.drop(tofilter.index[x], axis=0, ) #dropping
invalid rows

        ''' Formating beacons address'''


valid_rows[['Beacons','RSSI','bx','cx','xx']]=valid_rows[4].str.split('
',expand=True,)
        data_expand =valid_rows.dropna(axis=1)
        data=data_expand.drop(columns = [0, 4, 5,'bx','cx','xx'])
        data['Beacons'].mask(data['Beacons']=='c8:49:6d:cf:9a:b8', 'TAG_1',
inplace = True)
        data['Beacons'].mask(data['Beacons']=='d4:32:46:39:8e:a3', 'TAG_2',
inplace = True)
        data['Beacons'].mask(data['Beacons']=='d9:3c:4c:a1:ee:5a', 'TAG_3',
inplace = True)
        data['Beacons'].mask(data['Beacons']=='e8:a8:08:43:d0:25', 'TAG_4',
inplace = True)
        data['Beacons'].mask(data['Beacons']=='f4:30:c7:04:a2:44', 'TAG_5',
inplace = True)

        ''' Formating RSSI data '''

        data['RSSI']=data['RSSI'].str.replace(r'\W',"")
        data['RSSI']= -1*pd.to_numeric(data['RSSI'])

        a = data['Beacons'].unique()
        new_df = pd.DataFrame()
        for i in range(len(a)):
            proxy_mil = data[data["Beacons"]== a[i]]
            # print(proxy_mil)
            proxy_mil  = proxy_mil [proxy_mil ["RSSI"]==
proxy_mil.RSSI.max()]
            new_df = pd.concat([new_df, proxy_mil])


        readinglist = new_df.values.tolist()
        reading_data = pd.DataFrame(readinglist, columns=['Gate way
ID','TIME STAMP', 'MAC ADDRESS', 'EB_RSSI','Beacons','RSSI'])

        filtered_data = reading_data[columns1]

        good_data= len(filtered_data)
        zone1 = filtered_data[filtered_data['MAC ADDRESS']
=="58:8E:81:A5:FD:84"]
        zone2 = filtered_data[filtered_data['MAC ADDRESS']
=="58:8E:81:A5:FD:94"]

        EB1_records = zone1.values.tolist()
        EB2_records = zone2.values.tolist()
        call_to_SQL(EB1_records,EB2_records,good_data,new_data);

    else :
      print("no new data")

def call_to_SQL(recs1,recs2,recorded_rows,total_rows) :
     DRIVER = 'SQL Server'
     SERVER_NAME = 'DESKTOP-LNDLNE9'
     DATABASE_NAME = 'BLUEPYC'

     def connection_string( driver, server_name, database_name):
          conn_string = f"""
              DRIVER= {{{driver}}};
              SERVER={server_name};
              DATABASE={database_name};
              Trusted_connection=no
              UID=LaVida;
              PWD=*******;
          """
          return conn_string
     try:
          conn = odbc.connect(connection_string(DRIVER, SERVER_NAME,
DATABASE_NAME))

     except odbc.DatabaseError as e:
        print('Database Error:')
        print(str(e.value[1]))
     except odbc.Error as e:
        print('Connection Error:')
        print(str(e.value[1]))

     sql_insert_zone1 = '''
        INSERT INTO Zone1
        VALUES (?,?,?,?,?,GETDATE())
        '''
     sql_insert_zone2 = '''
        INSERT INTO Zone2
        VALUES (?,?,?,?,?,GETDATE())
        '''
     try:
        cursor= conn.cursor()

        if len(recs1)<1:
            cursor.commit()
        else:
            cursor.executemany(sql_insert_zone1, recs1)
        if len(recs2)<1:
            cursor.commit()
        else:
            cursor.executemany(sql_insert_zone2, recs2)

        cursor.commit();
     except Exception as e:
        cursor.rollback()
        print(str(e[1]))
     finally:
        print("database updated with", recorded_rows, "rows ",total_rows-
recorded_rows, "bad rows skipped")
        cursor.close()
        conn.close()

while True:
    df=pd.read_csv(r'C:\Program Files
(x86)\BluEpyc\BluEpycStart\Log_06_05_2021.txt',
                   sep=';',header=None)
    length=len(df)
    time.sleep(2)
    datafilter(length)
```

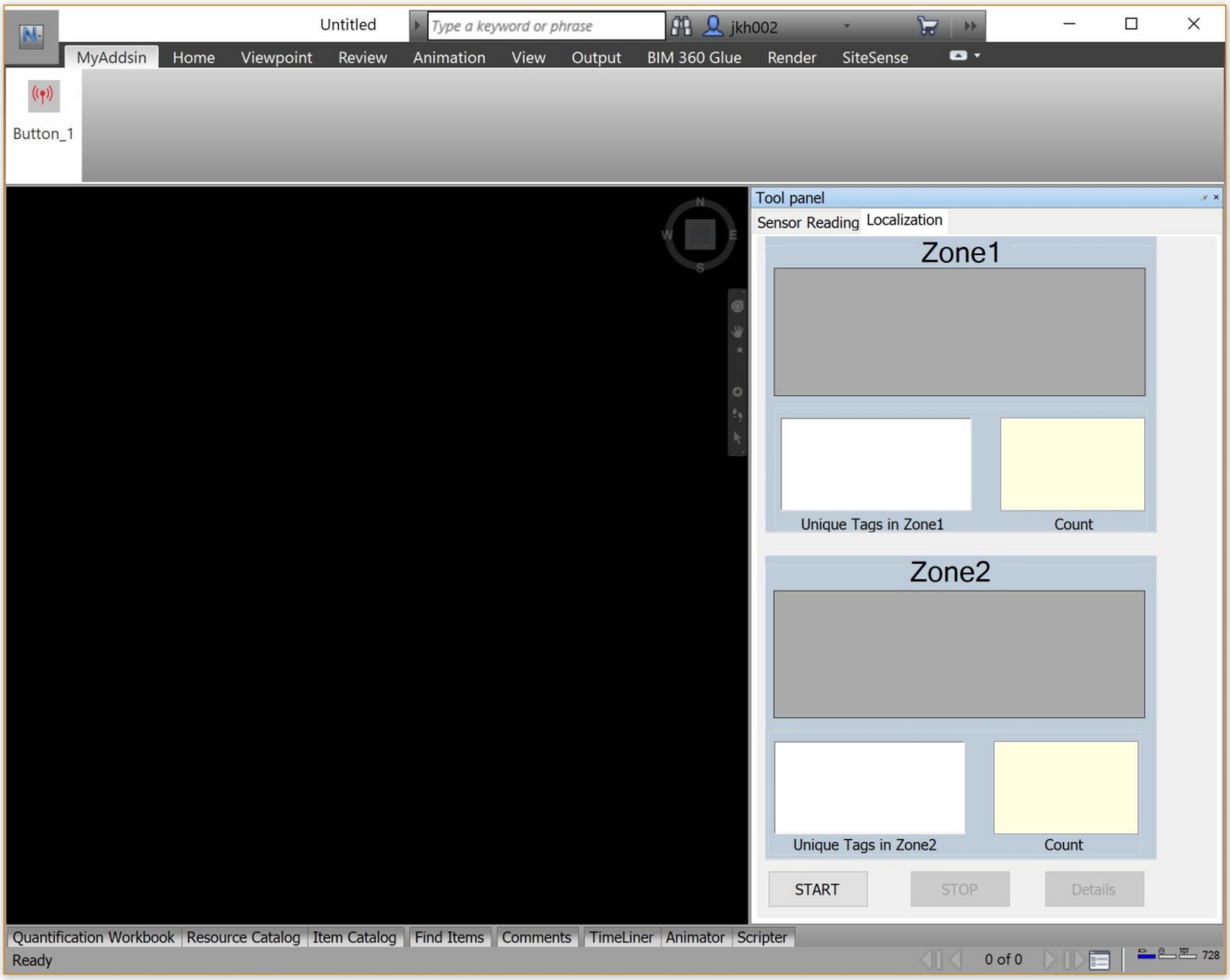
Appendix D

1. Program for Localization data visualizations.

```
using System;
using System.Data;
using System.Windows.Forms;
using Autodesk.Navisworks.Api;
using System.Data.SqlClient;
using System.Threading;
using Application = System.Windows.Forms.Application;
using App = Autodesk.Navisworks.Api.Application;
using System.Drawing;

namespace AddinRibbon.Ctr
{

    public partial class Locaization : UserControl
    {

        protected override void OnParentChanged(EventArgs e)
        {
            base.OnParentChanged(e);
            Dock = DockStyle.Fill;
        }

        Font fnt = new Font("Times New Roman", 32.0f, FontStyle.Italic);
        Document oDoc = App.ActiveDocument;

        DataTable Zone1 = new DataTable();
        DataTable Zone2 = new DataTable();

        Thread updateDataGridView;
        Thread updateDataGridView2;



        static System.Windows.Forms.Timer myTimer = new System.Windows.Forms.Timer();
        static bool exitFlag = false;

        int Z1 = 0;
        int Z2 = 0;
        int count1 = 0;
        int count2 = 0;
        int m1 =0;
        int m2 = 0;

        string S1 = "";
        string S2 = "";
        public Locaization()
        {
            InitializeComponent();
            Zone1.Columns.Add("Time Stamp");
            Zone1.Columns.Add("Tag");
            Zone1.Columns.Add("Sensor_no");
            Zone1.Columns.Add("RSSI");

            Zone2.Columns.Add("Time Stamp");
            Zone2.Columns.Add("Tag");
            Zone2.Columns.Add("Sensor_no");
            Zone2.Columns.Add("RSSI");
            myTimer.Tick += new EventHandler(TimerEventProcessor);
        }

        public string conn_str = "Data Source=DESKTOP-LNDLNE9;Initial 
Catalog=BLUEPYC;Integrated Security=True";

        #region Start_Button click
        private void View_btn_Click(object sender, EventArgs e)
        {

            Backup_data();
            try
            {

                if (App.ActiveDocument.FileName != "")
                {
                    ModelItem rootItem = oDoc.Models[0].RootItem;
                    if (rootItem.DisplayName == "Testroom2_V02.nwd")
                    {
                        exitFlag = false;
                        View_btn.Enabled = false;
                        Stop_btn.Enabled = true;
                        Details_btn.Enabled = false;


                        myTimer.Interval = 1000;
                        myTimer.Start();
                        while (exitFlag == false)
                        {
                            // Processes all the events in the queue.
                            Application.DoEvents();
                        }
                    }
                }
                else
                {
                    MessageBox.Show("Open Testroom2_V02.nwd", "Message", 
MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
            }
            catch (Exception c)
            {
                MessageBox.Show($"Message! {c}");
            }
        }
        #endregion

        #region Timer_Settings
        private void TimerEventProcessor(Object myObject,
                                                EventArgs myEventArgs)
        {
            myTimer.Stop();
            if (exitFlag != true)
            {

                ReadSQL_Data();

                myTimer.Enabled = true;
            }
        }
        #endregion


        #region Database readings
        public void ReadSQL_Data()
        {
            SqlConnection con = new SqlConnection(conn_str);
            con.Open();
            if (con.State == ConnectionState.Open)
            {
                using (var count_row_Z1 = con.CreateCommand())
                {
                    count_row_Z1.CommandText = @"select Count(*) from Zone1;";
                    int R1 = (int)count_row_Z1.ExecuteScalar();

                    if (R1 > Z1)
                    {
                        int Diff = Z1 - R1;
                        label3.Visible = false;
                        try
                        {
                            using (var command1 = con.CreateCommand())
                            {
                                command1.CommandText = @"Select *FROM Zone1 ORDER BY 
TIME_STAMP ASC offset @Z1 rows;";
                                command1.Parameters.Add(new SqlParameter("@Z1", Z1));
                                using (SqlDataReader read1 = command1.ExecuteReader())
                                {
                                    while (read1.Read())
                                    {
                                        DataRow dataRow = Zone1.NewRow();
                                        dataRow["Time Stamp"] = read1["TIME_STAMP"];
                                        dataRow["Sensor_no"] = read1["EB_ADDRESS"];
                                        dataRow["Tag"] = read1["TAG"];
                                        dataRow["RSSI"] = read1["RSSI"];
                                        Zone1.Rows.InsertAt(dataRow, 0);
                                        Visualize(dataRow);

                                         Z1 = Zone1.Rows.Count;
                                        m1++;
                                    }
                                    DataTable dtCopy = Zone1.Clone();
                                    if (Zone1.Rows.Count > 0)
                                    {
                                        for (int i = 0; i < m1; i++)
                                        {
                                            dtCopy.ImportRow(Zone1.Rows[i]);
                                        }
                                    }
                                    int Xrows =dtCopy.Rows.Count;
                                    DataView view = new DataView(dtCopy);
                                    DataTable distinct_tag = view.ToTable(true, 
"Tag");


                                    textBox1.Font = fnt;
                                    count1 = distinct_tag.Rows.Count;
                                    textBox1.Text = count1.ToString();
                                    richTextBox1.Clear();

                                    for (int i = 0; i < count1; i++)
                                    {
                                        richTextBox1.Text += Environment.NewLine +
distinct_tag.Rows[i]["Tag"].ToString();
                                    }

                                    distinct_tag.Rows.Clear();
                                    m1 = 0;
                                    dataGridView1.DataSource = Zone1;
                                    updateDataGridView = new Thread(new
ThreadStart(RealTimeData));

                                    updateDataGridView.Start();
                                    dataGridView1.FirstDisplayedScrollingRowIndex = 0;
                                    dataGridView1.Refresh();
                                }
                            }
                        }
                        catch (Exception e)
                        {
                            MessageBox.Show($"message {e}");
                        }
                    }
                    else
                    {
                        label3.Visible = true;
                        label3.Text = "No recent data to update in Zone1";

                        Reset("329137", "329391");
                        richTextBox1.Text =" ";
                        textBox1.Clear();

                    }


                }
                using (var count_row_Z2 = con.CreateCommand())
                {
                    count_row_Z2.CommandText = @"select Count(*) from Zone2;";
                    int R2 = (int)count_row_Z2.ExecuteScalar();
                    if (R2 > Z2)
                    {
                        label4.Visible = false;
                        try
                        {
                            using (var command2 = con.CreateCommand())
                            {
                                command2.CommandText = @"Select *FROM Zone2 ORDER BY
TIME_STAMP ASC offset @Z2 rows;";
                                command2.Parameters.Add(new SqlParameter("@Z2", Z2));
                                using (SqlDataReader read2 = command2.ExecuteReader())
                                {
                                    while (read2.Read())
                                    {
                                        DataRow dataRow = Zone2.NewRow();
                                        dataRow["Time Stamp"] = read2["TIME_STAMP"];
                                        dataRow["Sensor_no"] = read2["EB_ADDRESS"];
                                        dataRow["Tag"] = read2["TAG"];
                                        dataRow["RSSI"] = read2["RSSI"];
                                        Visualize(dataRow);
                                        Zone2.Rows.InsertAt(dataRow, 0);
                                        Visualize(dataRow);
```

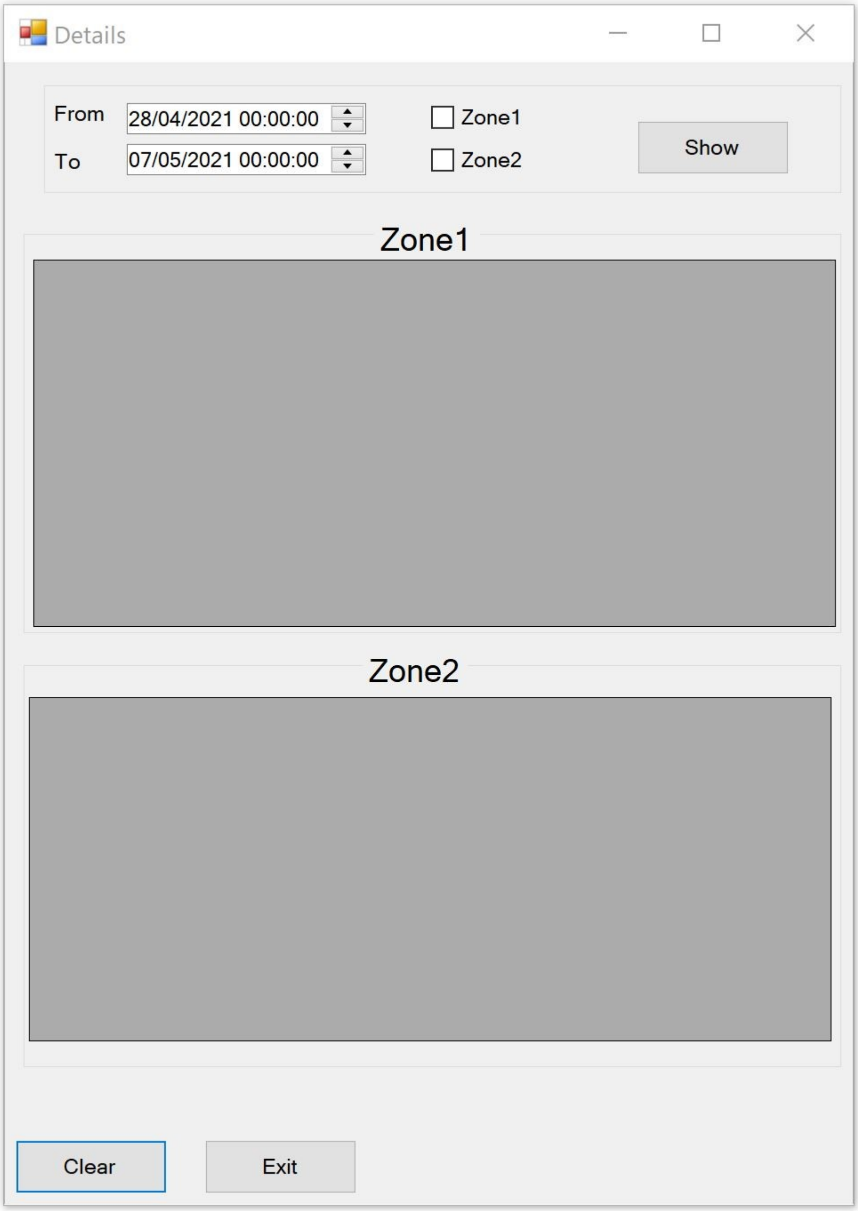
```
                                         Z2 = (int)Zone2.Rows.Count;
                                        m2++;
                                    }

                                    DataTable dtcopy2 = Zone2.Clone();

                                    if (Zone2.Rows.Count > 0)
                                    {
                                        for (int i = 0; i < m2; i++)
                                        {
                                            dtcopy2.ImportRow(Zone2.Rows[i]);
                                        }
                                    }
                                    int Xrows = dtcopy2.Rows.Count;
                                    DataView view2 = new DataView(dtcopy2);
                                    DataTable distinct_tag2 = view2.ToTable(true,
"Tag");

                                    textBox2.Font = fnt;
                                    count2 = distinct_tag2.Rows.Count;
                                    textBox2.Text = count2.ToString();
                                    richTextBox2.Clear();
                                    for (int i = 0; i < count2; i++)
                                        {
                                            richTextBox2.Text += Environment.NewLine +
distinct_tag2.Rows[i]["Tag"].ToString();
                                        }

                                    distinct_tag2.Rows.Clear();
                                    m2 = 0;
                                    dataGridView2.DataSource = Zone2;
                                    updateDataGridView2 = new Thread(new
ThreadStart(RealTimeData2));

                                    updateDataGridView2.Start();
                                    dataGridView2.FirstDisplayedScrollingRowIndex = 0;
                                    dataGridView2.Refresh();
                                }
                            }
                        }
                        catch (Exception e)
                        {
                            MessageBox.Show($"message {e}");
                        }
                    }
                    else
                    {
                        label4.Visible = true;
                        label4.Text = "No recent data to update in Zone2";

                        Reset("329038", "328771");
                        richTextBox2.Text =" ";
                        textBox2.Clear();

                    }

                   con.Close();
                }

            }

        }
        #endregion
        #region Naviswork visualization
        public void Visualize(DataRow row)
        {
            String Field = row["Sensor_no"].ToString();
            var T = new Search();
            T.Locations = SearchLocations.DescendantsAndSelf;
            T.Selection.SelectAll();

            SearchCondition Group1_Sc1 =
SearchCondition.HasPropertyByDisplayName("Element ID", "Value");
            SearchCondition Group1_Sc2 =

SearchCondition.HasPropertyByDisplayName("Element ID", "Value");


if (Field == "58:8E:81:A5:FD:84")
            {

                Group1_Sc1 =
Group1_Sc1.EqualValue(VariantData.FromDisplayString("329137"));
                Group1_Sc2 =
Group1_Sc2.EqualValue(VariantData.FromDisplayString("329391"));
                System.Collections.Generic.List<SearchCondition> oG1 = new
System.Collections.Generic.List<SearchCondition>();
                oG1.Add(Group1_Sc1);
                System.Collections.Generic.List<SearchCondition> oG2 = new
System.Collections.Generic.List<SearchCondition>();
                oG2.Add(Group1_Sc2);
                T.SearchConditions.AddGroup(oG1);
                T.SearchConditions.AddGroup(oG2);
                ModelItemCollection set1 = T.FindAll(oDoc, false);
                ModelItemCollection SearchResults = set1;
                oDoc.CurrentSelection.CopyFrom(SearchResults);
                oDoc.Models.OverrideTemporaryTransparency(SearchResults, 0.5);
                oDoc.Models.OverrideTemporaryColor(SearchResults, new
Autodesk.Navisworks.Api.Color(0, 1, 0));
                oDoc.CurrentSelection.Clear();
            }
            else if (Field =="58:8E:81:A5:FD:94")
            {
                Group1_Sc1 =
Group1_Sc1.EqualValue(VariantData.FromDisplayString("329038"));
                Group1_Sc2 =
Group1_Sc2.EqualValue(VariantData.FromDisplayString("328771"));
                System.Collections.Generic.List<SearchCondition> oG1 = new
System.Collections.Generic.List<SearchCondition>();
                oG1.Add(Group1_Sc1);
                System.Collections.Generic.List<SearchCondition> oG2 = new
System.Collections.Generic.List<SearchCondition>();
                oG2.Add(Group1_Sc2);
                T.SearchConditions.AddGroup(oG1);
                T.SearchConditions.AddGroup(oG2);
                ModelItemCollection set2 = T.FindAll(oDoc, false);
                ModelItemCollection SearchResults = set2;
                oDoc.CurrentSelection.CopyFrom(SearchResults);
                oDoc.Models.OverrideTemporaryTransparency(SearchResults, 0.5);
                oDoc.Models.OverrideTemporaryColor(SearchResults, new
Autodesk.Navisworks.Api.Color(0, 1, 0));
                oDoc.CurrentSelection.Clear();
            }

        }
        #endregion

        #region DataGrid Visualizations
        private void RealTimeData()
        {

            DataTable test_table = Zone1;
            SetDataSourceInGridView(test_table);
        }
        delegate void GridViewArgReturningVoidDelegate(DataTable test_table);
        private void SetDataSourceInGridView(DataTable test_table)
        {
            if (this.dataGridView1.InvokeRequired)
            {
                GridViewArgReturningVoidDelegate d = new
GridViewArgReturningVoidDelegate(SetDataSourceInGridView);
                this.Invoke(d, new Object[] { test_table });

            }
            else
            {
                this.dataGridView1.DataSource = test_table;
                this.dataGridView1.Update();
                this.dataGridView1.Refresh();
                //dataGridView1.FirstDisplayedScrollingRowIndex = 0;
            }
        }
        private void RealTimeData2()
        {
            DataTable test_table2 = Zone2;
            SetDataSourceInGridView2(test_table2);

        }
        delegate void GridViewArgReturningVoidDelegate2(DataTable test_table2);
        private void SetDataSourceInGridView2(DataTable test_table2)
        {
            if (dataGridView2.InvokeRequired)
            {
                GridViewArgReturningVoidDelegate2 e = new
GridViewArgReturningVoidDelegate2(SetDataSourceInGridView2);
                Invoke(e, new Object[] { test_table2 });

            }
            else
            {
                dataGridView2.DataSource = test_table2;
                dataGridView2.Update();
                dataGridView2.Refresh();

            }

        }
        #endregion

        #region Stop_button
        private void button1_Click(object sender, EventArgs e)
        {
            View_btn.Enabled = true;
            Stop_btn.Enabled = false;
            Details_btn.Enabled = true;
            try
            {
                exitFlag = true;
                //myTimer=null;
                oDoc.Models.ResetAllTemporaryMaterials();

                DataTable DT1 = (DataTable)dataGridView1.DataSource;
                if (DT1 != null)
                {
                    DT1.Clear();
                    dataGridView1.Refresh();
                }
                DataTable DT2 = (DataTable)dataGridView2.DataSource;
                if (DT2 != null)
                {
                    DT2.Clear();
                    dataGridView2.Refresh();
                }

                label3.Text = "Stopped";
                label3.Visible = true;
                label4.Visible = false;
                Backup_data();
                Zone1.Clear();
                Zone2.Clear();
                Z1 = (int)Zone1.Rows.Count;
                Z2 = (int)Zone2.Rows.Count;
            }

            catch (Exception c)
            {
                MessageBox.Show($"message {c}");
            }

        }
        #endregion

        #region Clean ups

        private void Reset(String F, String S)
        {
            var T = new Search();
            T.Locations = SearchLocations.DescendantsAndSelf;
            T.Selection.SelectAll();

            SearchCondition Group1_Sc1 = 
SearchCondition.HasPropertyByDisplayName("Element ID", "Value");
            Group1_Sc1 = Group1_Sc1.EqualValue(VariantData.FromDisplayString(F));
            SearchCondition Group1_Sc2 = 
SearchCondition.HasPropertyByDisplayName("Element ID", "Value");
            Group1_Sc2 = Group1_Sc2.EqualValue(VariantData.FromDisplayString(S));
            System.Collections.Generic.List<SearchCondition> oG1 = new 
System.Collections.Generic.List<SearchCondition>();
            oG1.Add(Group1_Sc1);
            System.Collections.Generic.List<SearchCondition> oG2 = new 
System.Collections.Generic.List<SearchCondition>();
            oG2.Add(Group1_Sc2);
            T.SearchConditions.AddGroup(oG1);
            T.SearchConditions.AddGroup(oG2);

            ModelItemCollection set1 = T.FindAll(oDoc, false);
            oDoc.Models.ResetTemporaryMaterials(set1);
        }

        private void Backup_data()
        {
            SqlConnection con = new SqlConnection(conn_str);
            con.Open();
            if (con.State == System.Data.ConnectionState.Open)
            {
                using (var Backup1 = con.CreateCommand())
                {
                    Backup1.CommandText = @"Insert INTO Backup1 select *from  Zone1;";
                    Backup1.ExecuteNonQuery();
                }

                using (var Backup2 = con.CreateCommand())
                {
                    Backup2.CommandText = @"insert INTO Backup2 select *from Zone2;";
                    Backup2.ExecuteNonQuery();
                }

                using (var Empty1 = con.CreateCommand())
                {
                    Empty1.CommandText = @"Delete from Zone1;";
                    Empty1.ExecuteNonQuery();
                }

                using (var Empty2 = con.CreateCommand())
                {
                    Empty2.CommandText = @"Delete from Zone2;";
                    Empty2.ExecuteNonQuery();
                }

            }

            #endregion

        }

        private void button1_Click_1(object sender, EventArgs e)
        {
            new Details().Show();
            Details_btn.Enabled = false;
        }
    }
}
```

2. *Developed tool panels for Localization data visualizations.*



Figur iii Tool panel for visualization of Localizataion data

3. *Tool panel to access historical Localization data visualizations.*



From 28/04/2021 00:00:00

To 07/05/2021 00:00:00

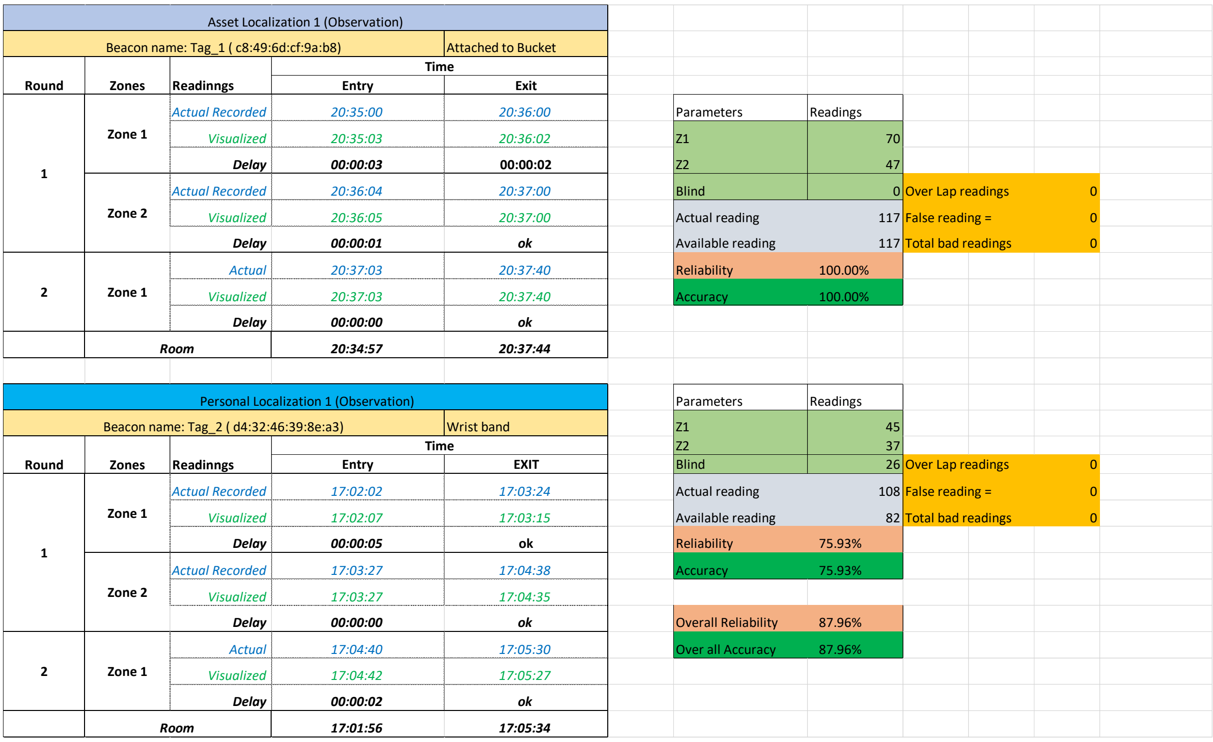
☐ Zone1

☐ Zone2

Figur iv Tool panel for accesing historical data for localizations (when 'Details' button pressed)

Appendix E

1. *Over all calculations for. Single Target (Assets & Personnel)*



Asset Localization 1 (Observation)				
Beacon name: Tag_1 (c8:49:6d:cf:9a:b8)				Attached to Bucket
			Time	
Round	Zones	Readinngs	Entry	Exit
1	Zone 1	*Actual Recorded*	*20:35:00*	*20:36:00*
		Visualized	*20:35:03*	*20:36:02*
		Delay	***00:00:03***	**00:00:02**
	Zone 2	*Actual Recorded*	*20:36:04*	*20:37:00*
		Visualized	*20:36:05*	*20:37:00*
		Delay	***00:00:01***	***ok***
2	Zone 1	*Actual*	*20:37:03*	*20:37:40*
		Visualized	*20:37:03*	*20:37:40*
		Delay	***00:00:00***	***ok***
	Room		***20:34:57***	***20:37:44***

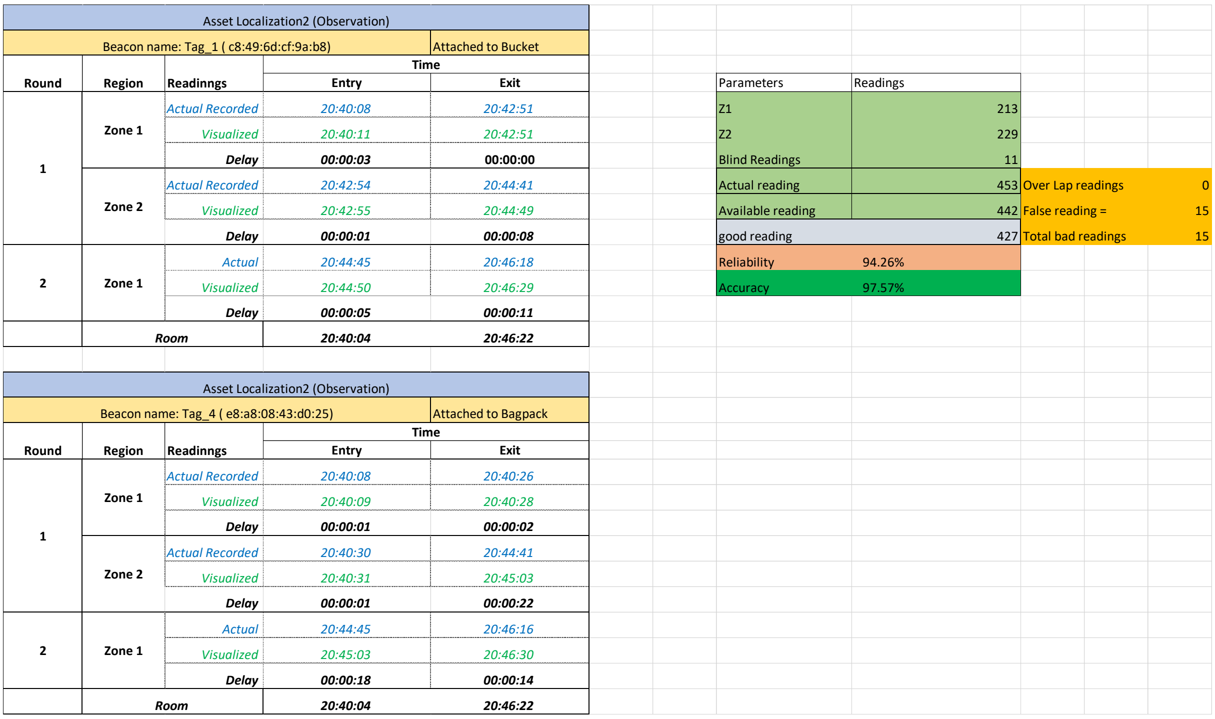
Parameters	Readings		
Z1	70		
Z2	47		
Blind	0	Over Lap readings	0
Actual reading	117	False reading =	0
Available reading	117	Total bad readings	0
Reliability	100.00%		
Accuracy	100.00%		

Personal Localization 1 (Observation)				
Beacon name: Tag_2 (d4:32:46:39:8e:a3)				Wrist band
			Time	
Round	Zones	Readinngs	Entry	EXIT
1	Zone 1	*Actual Recorded*	*17:02:02*	*17:03:24*
		Visualized	*17:02:07*	*17:03:15*
		Delay	***00:00:05***	**ok**
	Zone 2	*Actual Recorded*	*17:03:27*	*17:04:38*
		Visualized	*17:03:27*	*17:04:35*
		Delay	***00:00:00***	***ok***
2	Zone 1	*Actual*	*17:04:40*	*17:05:30*
		Visualized	*17:04:42*	*17:05:27*
		Delay	***00:00:02***	***ok***
	Room		***17:01:56***	***17:05:34***

Parameters	Readings		
Z1	45		
Z2	37		
Blind	26	Over Lap readings	0
Actual reading	108	False reading =	0
Available reading	82	Total bad readings	0
Reliability	75.93%		
Accuracy	75.93%		
Overall Reliability	87.96%		
Over all Accuracy	87.96%		

Figure v: Observation table and calculations for single target localization (both Assets & Personnel)

2. *calculations for. double Target (Assets)*



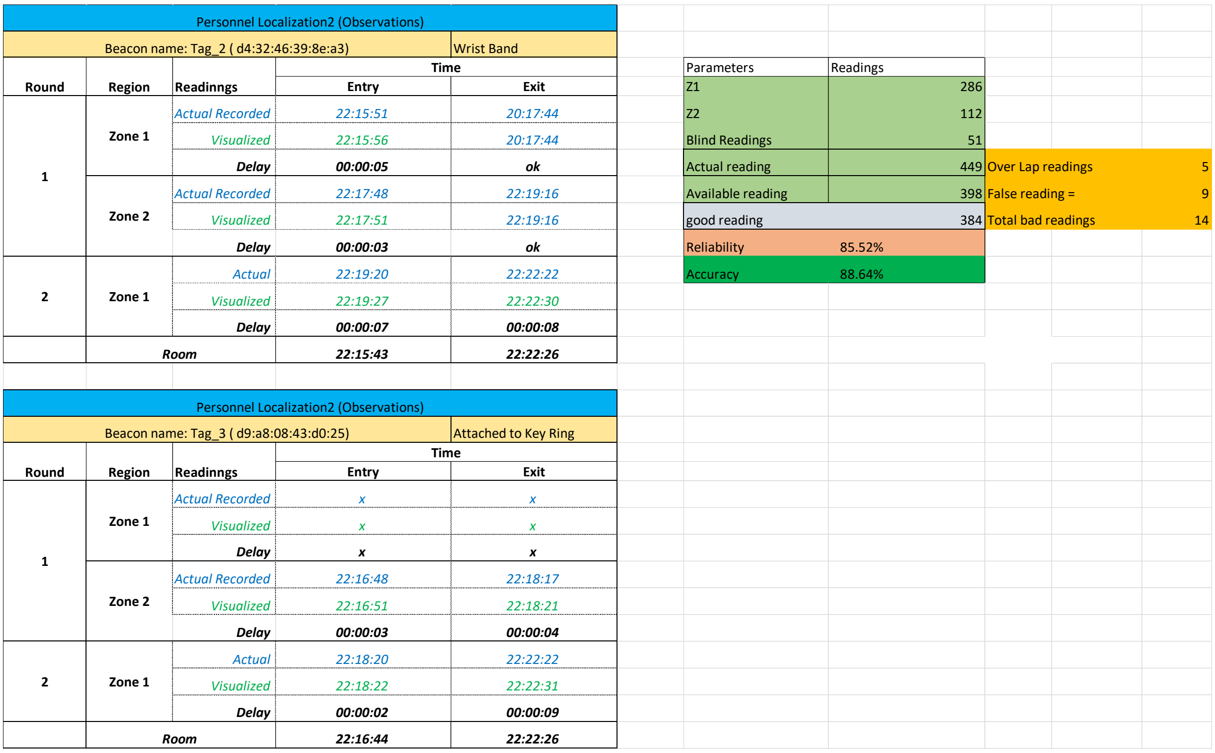
Asset Localization2 (Observation)				
Beacon name: Tag_1 (c8:49:6d:cf:9a:b8)				Attached to Bucket
			Time	
Round	Region	Readinngs	Entry	Exit
1	Zone 1	*Actual Recorded*	*20:40:08*	*20:42:51*
		Visualized	*20:40:11*	*20:42:51*
		Delay	***00:00:03***	**00:00:00**
	Zone 2	*Actual Recorded*	*20:42:54*	*20:44:41*
		Visualized	*20:42:55*	*20:44:49*
		Delay	***00:00:01***	***00:00:08***
2	Zone 1	*Actual*	*20:44:45*	*20:46:18*
		Visualized	*20:44:50*	*20:46:29*
		Delay	***00:00:05***	***00:00:11***
	Room		***20:40:04***	***20:46:22***

Parameters	Readings		
Z1	213		
Z2	229		
Blind Readings	11		
Actual reading	453	Over Lap readings	0
Available reading	442	False reading =	15
good reading	427	Total bad readings	15
Reliability	94.26%		
Accuracy	97.57%		

Asset Localization2 (Observation)				
Beacon name: Tag_4 (e8:a8:08:43:d0:25)				Attached to Bagpack
			Time	
Round	Region	Readinngs	Entry	Exit
1	Zone 1	*Actual Recorded*	*20:40:08*	*20:40:26*
		Visualized	*20:40:09*	*20:40:28*
		Delay	***00:00:01***	***00:00:02***
	Zone 2	*Actual Recorded*	*20:40:30*	*20:44:41*
		Visualized	*20:40:31*	*20:45:03*
		Delay	***00:00:01***	***00:00:22***
2	Zone 1	*Actual*	*20:44:45*	*20:46:16*
		Visualized	*20:45:03*	*20:46:30*
		Delay	***00:00:18***	***00:00:14***
	Room		***20:40:04***	***20:46:22***

Figure vi Observation table and calculations for single target localization (Assets)

3. *calculations for. double Target (Personnel)*



Personnel Localization2 (Observations)

Beacon name: Tag_2 (d4:32:46:39:8e:a3) | Wrist Band

Round	Region	Readinngs	Time Entry	 Exit
1	Zone 1	Actual Recorded	22:15:51	20:17:44
		Visualized	22:15:56	20:17:44
		Delay	00:00:05	ok
	Zone 2	Actual Recorded	22:17:48	22:19:16
		Visualized	22:17:51	22:19:16
		Delay	00:00:03	ok
2	Zone 1	Actual	22:19:20	22:22:22
		Visualized	22:19:27	22:22:30
		Delay	00:00:07	00:00:08
	Room		22:15:43	22:22:26

Parameters	Readings		
Z1	286		
Z2	112		
Blind Readings	51		
Actual reading	449	Over Lap readings	5
Available reading	398	False reading =	9
good reading	384	Total bad readings	14
Reliability	85.52%		
Accuracy	88.64%		

Personnel Localization2 (Observations)

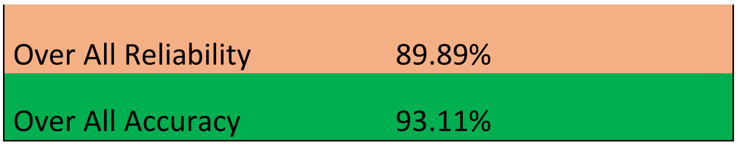
Beacon name: Tag_3 (d9:a8:08:43:d0:25) | Attached to Key Ring

Round	Region	Readinngs	Time Entry	 Exit
1	Zone 1	Actual Recorded	x	x
		Visualized	x	x
		Delay	x	x
	Zone 2	Actual Recorded	22:16:48	22:18:17
		Visualized	22:16:51	22:18:21
		Delay	00:00:03	00:00:04
2	Zone 1	Actual	22:18:20	22:22:22
		Visualized	22:18:22	22:22:31
		Delay	00:00:02	00:00:09
	Room		22:16:44	22:22:26

Figure vii Observation table and calculation for double target localization(Personnel)

4. *Over all calculations for double Target (Assets & Personnel)*

The overall reliability and accuracy for double target Localization was calculated to be followings:



Over All Reliability	89.89%
Over All Accuracy	93.11%

References & useful links:

[1] Navisworks API Training Lab Autodesk Community, 2016 http://images.autodesk.com/adsk/files/Navisworks_Training_Labs_2015_0921.zip

[2] “Navisworks API”, YouTube, uploaded by Connessioni Training, 2017

https://www.youtube.com/watch?v=UT4wvB6phSM&list=PLmNgmrUVbrOu3gTUkACQKBEkZEXpZz7Kb

[3] Kavgic, Miroslava, and Rasit Eskicioglu. "INTEGRATING BUILDING INFORMATION MODELING (BIM) AND SENSOR TECHNOLOGY FOR FACILITY MANAGEMENT." (2019).

[4]https://github.com/ControlEverythingCommunity/MCP9808/blob/master/Python/MCP9808.py

[5] https://github.com/jimbobbennett/Raspberry-Pi-And-Azure-IoTHub/blob/master/sensor.py

[6] https://www.instructables.com/Raspberry-PI-Multiple-I2c-Devices/