Smart Meeting Room Management System Based on Real-Time Occupancy

Luis Miguel Sánchez
University of Costa Rica
San Jose, Costa Rica
luis.sanchezvargas@ucr.ac.cr

Ignacio Díaz–Oreiro
University of Costa Rica
San Jose, Costa Rica
ignacio.diazoreiro@ucr.ac.cr

Luis Quesada
University of Costa Rica
San Jose, Costa Rica
luis.quesada@ecci.ucr.ac.cr

Luis A. Guerrero
University of Costa Rica
San Jose, Costa Rica
luis.guerrero@ecci.ucr.ac.cr

Gustavo López
University of Costa Rica
San Jose, Costa Rica
gustavo.lopez_h@ucr.ac.cr

***Abstract*— This paper proposes the creation of a smart meeting room through the incorporation of a PIR sensor and an AWS IoT button that allows the booking system to reflect a more precise availability of meeting rooms according to the actual occupancy status. The Internet of Things (IoT) devices are controlled using a Wi-Fi module that allows them to connect to the REST web service and to integrate with the open source Meeting Room Booking System (MRBS). In order to evaluate the system a storyboard evaluation was conducted with 47 participants. All participants filled out the User Experience Questionnaires (UEQ), described the product using three words and expressed their opinion through open comments. Finally, 19 participants took part in a real-life simulation of the smart meeting room and evaluated the system using the UEQ questionnaire. Based on the positive acceptance reflected in the evaluations, results show that the proposed system is considered very attractive and useful by the participants.**

Keywords*— *Internet of Things (IoT), Ambient Intelligence, Smart Meeting Room, Room Occupancy, User Experience, Smart Environments

I. Introduction

Meetings take place when two or more people need to coordinate or agree on something formally arranged [1]. Meetings are essential events in any organization as they are used to share knowledge and information or discuss and take important decisions [2]. In a more precise definition, a meeting is an activity that has a start and end time previously defined and is scheduled in the agenda of the attendees [3]. Statistical studies show that office workers spend around 30-70 percent of their daily work time in meetings [4].

According to [5], the duration of a meeting cannot be determined exactly in advance. Even if having a well-defined agenda, meetings can be over before scheduled time, for example, if there were no unexpected interruptions and the meeting flowed very quickly.

Even if the meeting organizer is confident that the meeting will take less time than expected, he or she might still book a period longer than required for the meeting, as a preventive measure. In this context, in [5], the following patterns were reported when scheduling meetings:

- Meeting is over before the schedule time.
- Meeting does not occur at all.
- Other resources required may not be available, for example a projector or a computer, and this prevents the meeting from taking place.

Even though the situations mentioned above, there is no feedback mechanism to inform users if the meeting actually took place or not. For instance, it is possible that although a conference room has been reserved in advance, it is left unused due to a last-minute cancellation. Therefore, if another user wants to reserve the conference room at that time, he or she would not have been able to, since room availability was not appropriately handled by the system.

To address problems like the one mentioned above, in this paper a smart meeting room management system is proposed, which provides real-time support to attendees through a simple access to the room state and real-time occupancy status. According to [6] a smart meeting room (SMR) can be monitored by several sensors to collect information in real time such as temperature, humidity, and detection of people. In our proposal, we integrate a PIR sensor for movement detection, a microcontroller to monitor the sensor's behavior and to connect to internet, an actuator that allows users to make a room available if they conclude the meeting before time, a real-time occupancy status display module, and finally a web API service. Additionally, as part of the system architecture, the indicated system is integrated with the open source MRBS. In addition, to evaluate the system two different user experience studies were applied, one running a storyboard and another having participants interacting with the actual system. In both studies, the data was collected using the UEQ standardized questionnaire for UX evaluation.

The paper is organized as follows: section II briefly discusses the related work, section III describes the proposed

This work was partially supported by *Centro de Investigaciones en Tecnologías de la Información y Comunicación* (CITIC), *Escuela de Ciencias de la Computación e Informática* (ECCI) both at Universidad de Costa Rica. Grant No. 326-B9-107.

SRM system, section IV shows evaluation process and results, and finally section V shows conclusions and future work.

II. Related Work

Most of the research found regarding meeting room systems focuses on improving programming software to help users to select optimal meeting time [7] [8], while others build smart meeting rooms where audiovisual content is automatically recorded for viewing in the future [9] [10] [11]. However, these systems do not address the problem of making available meeting rooms that have been reserved but are not really occupied. In other words, users can not identify the actual state of a meeting room, since these rooms do not detect the presence or absence of people and therefore cannot warn that a room is no longer being used.

There are currently some meeting room applications [12] [13] [14] that have the functionality to make reservations and show the availability of the rooms through an interactive screen placed outside each room. They provide synchronization with booking systems like Microsoft Outlook and Google Calendar. However, these systems are not free of cost, and do not correctly handle room's availability, since all booking mechanisms are operated manually by users. Another related system is described in [5] which is based on a wireless sensor device connected to a mesh network. The occupancy status is not integrated into the booking application as it uses Microsoft Outlook software. The system needs manual action to figure out a room's underutilization. Another work worth mentioning is found in [2]. The proposed system is using PIR fusion modules which was the combination of three or four PIR sensors integrated with its own management system. However, this system used embedded modules to connect to LAN through existing Ethernet cable.

In contrast to all approaches mentioned, the proposed architecture provides a real-time occupancy status display module, an AWS IoT button that acts as an actuator, a Wi-Fi module to connect to the internet, and a web API service deployed to the cloud. Moreover, the proposed architecture is also scalable, since any other device can be added with the minimal effort. The modules mentioned above, and the system architecture is described in next section.

III. Proposed System

In this section, the design and development process of the SMR system and its implementation is explained. This section is organized as follows: subsection A describes management application and real time occupancy module, subsection B explain how the system is integrated with MRBS, subsection C shows the design and development process of the wireless module, subsection D lists room policies, subsection E explain the implementation of the AWS IoT button, and finally subsection F shows the steps to deploy the SMR system to a meeting room.

A. System overview

The management system is a web-based application hosted in the cloud. Due to its responsive design, it is a cross–platform application. From this application, users can manage booking, see a room's availability, search available rooms and get reminders.

An external module to have real-time access to the occupancy status of a room is located outside every room. Also, users have access to see all booked meetings during the day. This module is described in Fig. 1.

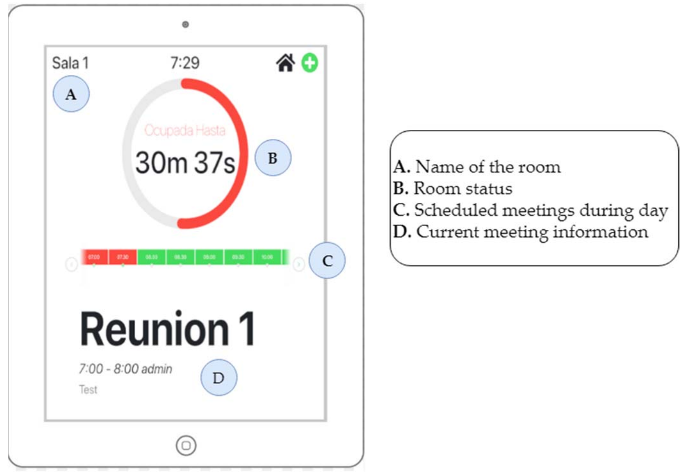


Fig. 1. Real-Time Occupancy Status Interface.

For the back-end layer, a microservice architecture was implemented [15]. Then, in case the system grows, additional have small services running on their own processes and communicating with each other through a lightweight mechanism. A Web API service was built using REST protocol. This service is the one in charge of allowing the entire system to communicate with itself; it receives requests from clients and sends response messages to them. Additionally, all important data is retrieved and stored in a MySQL database. The service is deployed to the cloud, so it could be used by any client that has a valid token. Fig. 2 presents a diagram of the system architecture. It can be seen that the Web API service is the most important module of the system.

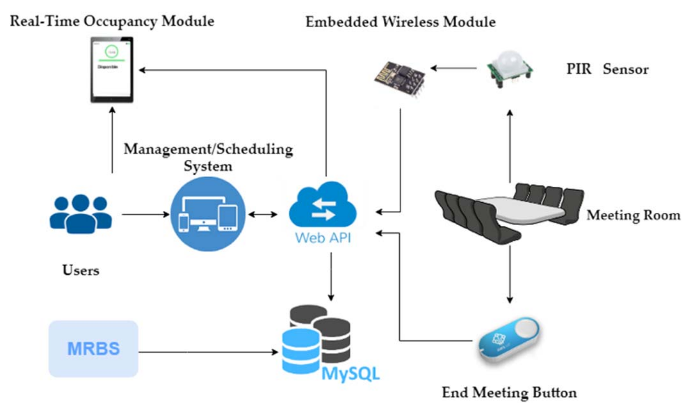


Fig. 2. System Archiquecture.

B. Synchronization with the MRBS

The proposed system has the ability to work as a standalone version or to work together with the Meeting Room Booking System (MRBS). The MRBS is an open source system used to manage meeting rooms. MRBS is a web application based on PHP with support for MySQL database. MRBS allows users to make reservations for a specific room at a specific time, checking the availability of the room before creating the reservation. According to [16], the system is currently in use by many large organizations around the world, which prompted us to use it to incorporate the integration with sensors and modules proposed in this paper.

This integration occurs in the Web API, as shown in Fig. 2. Since MRBS is an open source, and do not provide a public API, the same database structure to build the system was preferred. This means that, when both systems are synchronized, they will use the same database.

C. *Wireless transmission*

An embedded programming module is used in the system for processing the inputs and outputs from a given meeting room. The ESP8266 ESP-01 is a Wi-Fi module that allows microcontrollers access to a Wi-Fi network. This module is self-contained and can be programed to act as microcontroller as well. It has two GPIO [17] pins: GPIO0 and GPIO2 which allows to have some devices connected to it (in this case a PIR motion sensor [18] was connected to a GPIO2 pin). When the PIR motion sensor detects human presence, it will send high inputs to the module; otherwise it will be a low input. The ESP module was programed via Arduino UNO [19], using the Arduino IDE. Arduino has its own programming language, which essentially is C++ with some additional methods and functions. After uploading the code to the Wi-Fi module, the Arduino board was removed and replaced by a small breadboard power supply module of 3.3V, as shown in Fig. 3.

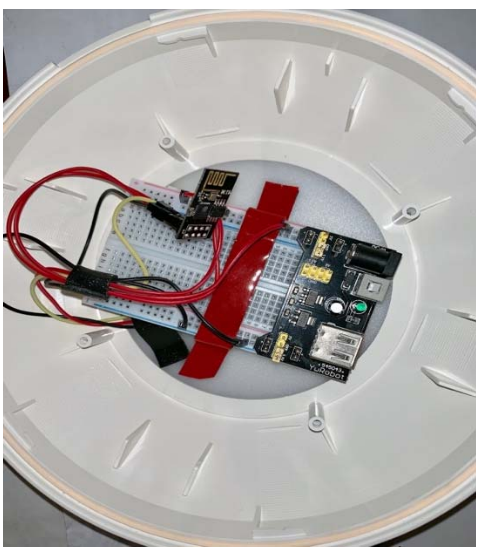


Fig. 3. Wi-Fi module.

Once the module is connected to the internet via Wi-Fi, a service account is used to get a valid token from the Web API. Then, the module is constantly asking to the web service if the room is currently reserved. If the web service responds true, the PIR sensor starts sending input signals to the module. Then, the system might be able to release a room based on room policies. This is explained in detail in subsection D.

D. *Room Policy*

Two rules related to the motion sensor were defined to determine if a meeting really starts. These rules are:

1) As of the start of the scheduled meeting time and if no movement is detected in the room for 15 minutes, the system will release the room and send a notification to the attendees. This initial time is configurable and can be changed by users, which mean it could change depending on user's needs.

2) If the first rule is not met and the system continues detecting movement after 15 minutes, the meeting is considered as consolidated. After this, the room only could be released by the actuator placed inside on it, this is explained in subsection E.

Based on these two rules, the system can determine if the participants presented themselves or not to the meeting, and if they did not show up, the room is released in the system so that it can be used by other people.

E. *End Meeting Button*

The system also uses an AWS IoT Button [20]. This is a programmable Wi-Fi button (see Fig. 4a) that allows participants to put the room available in case a meeting ends earlier than scheduled. The system implements an AWS Lambda function which is called once the user clicks on the AWS Button. Basically, a valid token from the web service is taken and then the system releases the room so it can be used by someone else.

F. *Room Setup*

Deploying the system to a room consists in placing the wireless module on the ceiling of the room (Fig. 4b), setting the AWS IoT button at the exit of the room (Fig. 4a) and finally setting the real-time occupancy status module at the entrance of the room.

(a) (b)

Fig. 4. Some Smart Meeting Room Management System modules. (a) AWS IoT button; (b) Wi-Fi module deployed in the ceiling of a room.

IV. EVALUATION PROCESS

To evaluate the SMR system, two different user experience studies were applied and are described below.

A. Storyboards

Before initiating the development of the system, it was decided to demonstrate to potential users some scenarios where they could interact with the most important features of the system. In order to evaluate this and get feedback from end users, a storyboard technique was conducted. Basically, a storyboard script which included scenarios to test, based on room occupancy availability.

The storyboard consisted as follows: a professor is walking around a meeting room. Since every room has its own current status display placed outside of the room, he notices that the room is available all day. He decides to book the room twice that day, once in the morning and once in the afternoon, for one hour in both cases. When the first meeting starts, the status of the room changes to busy, therefore it is not available to be reserved. Since the meeting ends 30 minutes earlier than expected, the meeting organizer proceeds to release the room through a button located at the exit of the room. Then the status of the room changes to available and can be booked by any other person in the system.

During that same day, the professor gets an emergency call from his wife, and as a result, he has to leave and he will not be able to attend the second meeting. He notifies the attendees, but he forgets to cancel the meeting in the system, therefore the meeting room is still reserved. When it is time to start the second meeting, the status of the room changes to busy again. At the same time, two other persons are looking for a meeting room for a last-minute meeting. They walk around the room and realize the room status is busy, but actually the room is empty. One of them continues looking for a room in the system and suddenly, notices that the same room that was busy a few minutes ago, was listed on the search result as available and ready to be booked. Then, the meeting room is reserved for 30 minutes. When these two people meet for the meeting, the meeting organizer asks how could he have booked the room if it was busy 15 minutes ago? The other person explains to him that the system releases the room after 15 minutes, if no motion is detected.

The storyboard described above was conducted in three different groups, and at the end of every presentation a user experience evaluation was performed. Participants were asked to fill out the UEQ questionnaire. A total of 47 participants completed the evaluation, 80 percent were male, and 20 percent were female. The participant's average age was 21 years.

The standardized UX questionnaire UEQ [21] provides an excel data analysis tool for capturing the experience of users, after they interacted with the product or system. It consists of six-dimension scales, including pragmatic and hedonic aspects. These dimensions are: Attractiveness: Do users like or dislike the product? Perspicuity: Is it easy to get familiar with the product and learn how to use it? (pragmatic). Efficiency: Can users solve their tasks without unnecessary effort? (pragmatic). Dependability: Does the user feel in control of the interaction? (hedonic). Stimulation: Is it exciting and motivating to use the product? (hedonic). Novelty: Is the product innovative and creative? (hedonic). UEQ considers scores starting at 0.8 as positive and starting at −0.8 as negative.

The means (the scale ranges from −3 horribly bad to +3 extremely good) and variance (in parenthesis) of the UEQ dimensions were: *Attractiveness* 1.41 (0.59), *Perspicuity* 1.89 (0.93), *Efficiency* 1.27 (0.61), *Dependability* 1.12 (0.58), *Stimulation* 0.45 (1.90) and *Novelty* 0.62 (1.13). According to these results participants were equally satisfied with the judgment of attractiveness and pragmatic dimensions and slightly less satisfied with the hedonic dimension. For testing reliability of dimensions, we calculated Cronbach's Alpha value for each dimension. The value obtained were: *Attractiveness,* 0.79; *Perspicuity,* 0.84; *Efficiency,* 0.93; *Dependability,* 0.92; *Stimulation,* 0.95; and *Novelty* 0.83. These values show that the reliability is in satisfactory ranges.

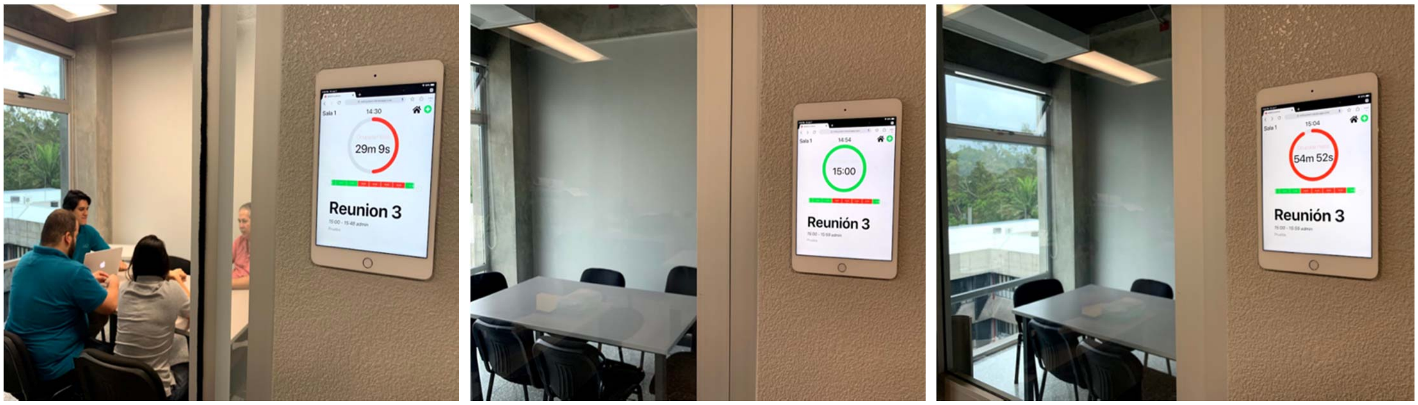
In addition, to the UEQ, participants were asked to describe the product in three words and allowed them to enter open comments related to the system. Most input from participants was that the product is very useful (Fig. 5). Moreover, several positive comments like *"It is a good idea; it can be useful"* or *"it could be a powerful tool in business"* were received. Some participants were concerned about a couple of scenarios were not covered by the storyboards, but also, they suggest some great ideas regarding how those scenarios can be addressed.

Fig. 5. Word cloud created using participants input. Each participant could use three words.

B. System evaluation

All the feedback collected in first evaluation was considered while the system was developed. Once the functional was stable, it was decided to run the same script described in the storyboard evaluation, but this time in a real scenario, allowing participants to interact with the product.

The SMR system was installed in a room (Fig. 6a, 6b, 6c), and then 19 participants were organized in groups of four. Before a group started interacting with the system, all their questions and doubts were answered, and then they were asked to execute equivalent actions as those explained for the storyboard.



(a) (b) (c)

Fig. 6. The real-time occupancy module displaying room status. (a) Room is currently busy. (b) Room is available for booking. (c) Room is busy but no attendees arrived

At the end of every iteration, participants were asked to fill out the UEQ questionnaire as participants did for the storyboard evaluation. Fig. 7 and Fig. 8 show all six dimensions have quite good results and participants were extremely satisfied with attractiveness, pragmatic, and hedonic aspect of the SMR application.

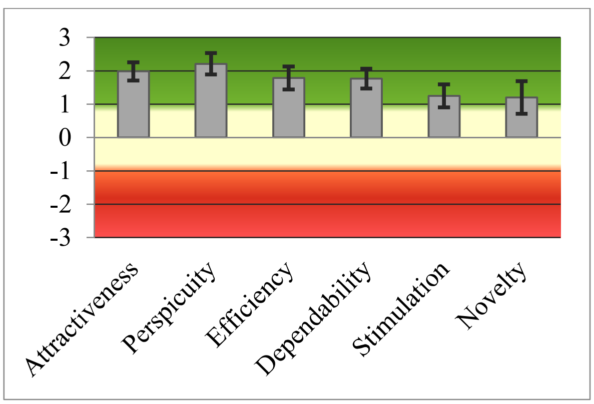


Fig. 7. The UEQ resultant scores or six dimensions scales.

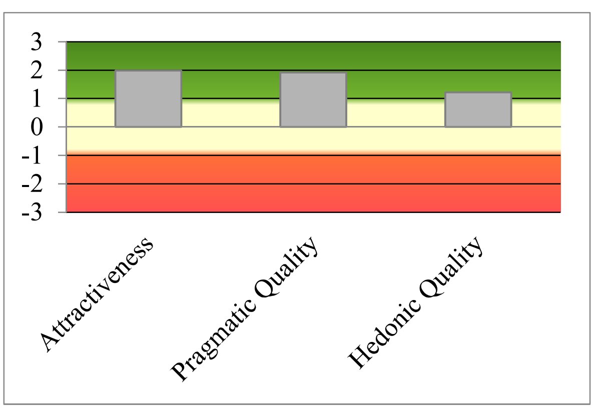


Fig. 8. The UEQ attractiveness, pragmatic and hedonic quality score.

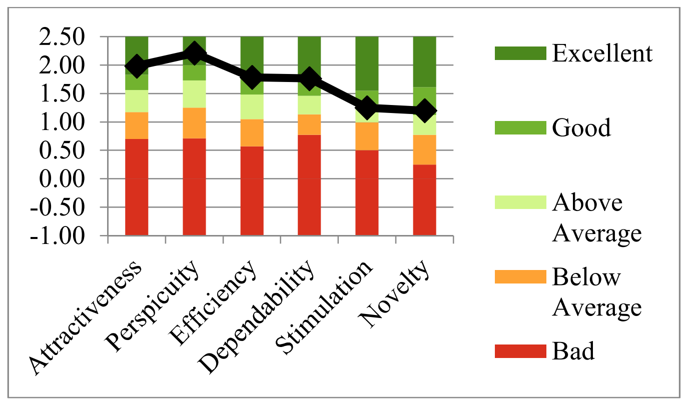


Fig. 9. UEQ resultants scores for six dimensions scales with benchmark data.

UEQ questionnaire [21] also provides a benchmark data set that contains data collected from 18,483 people from 401 studies of different products. The benchmark shows a comparative analysis of how good the evaluated product is, compared to the products in the benchmark dataset. Fig. 9 presents the UEQ results compared to those of 401 studies. Compared to the benchmark data, the evaluated SMR proposed in this paper scored between good and excellent, scoring in the top 10% best results for Attractiveness, Perspicuity, Efficiency and Dependability dimensions.

V. Conclusion and Future Work

We have created a real smart meeting room system that improves the visibility of meetings room availability by adding a PIR motion sensor and an AWS IoT button. We realized that one PIR sensor was enough for small meeting rooms. For large rooms we would probably have to use more than one sensor to cover all the room area.

Furthermore, using a wireless connection, as we implemented, is more efficient and less expensive than a wired connection proposed in other approaches. Also, developing a web service gives the flexibility to perform any kind of modification or improvement in the behavior of the IoTs devices, and provides the opportunity to create new services.

Concerning to the evaluation of UX, the results of the UEQ questionnaires show a positive acceptance by the users, both for the participants who used the storyboards and for those who did the test in the simulated environment. As expected, the evaluations of the simulated environment showed superior results to those of the storyboards, specifically in the *Stimulation* and *Novelty* dimensions. This is due to the differences between both environments, where the interaction with the actual sensors and devices presents the interaction with the system in a much more complete and interesting way. Additionally, if we compare the results with the general benchmarks provided by UEQ, the system shows a satisfactory user experience that makes us believe that the implementation of the proposed product in the real world would be highly successful.

In regards to future work, there are other scenarios not covered with the proposed system. However, we have collected some important input from users, thus, we can integrate these features later into the system. For instance, if the system stores sensor information, this information could go through machine learning algorithms and make the system take decision based on those algorithms. We can also make the system more complex by adding sensors for temperature and humidity, for example, and combine the information provided by these sensors with those of movement detection. In this way, if a sensor detects movements inside the room, which means a meeting is in progress, the system could perform operations in the microcontroller to bring in the preferred ambience by controlling appliances like the air conditioner. On the other hand, if the system detects that the room is empty, it could turn off air conditioners and save energy consumption.

REFERENCES

[1] M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning, and L. Suchman, “Beyond the chalkboard: computer support for collaboration and problem solving in meetings,” *Commun. ACM*, vol. 30, no. 1, pp. 32–47, Jan. 1987.

[2] L. D. Tran *et al.*, “A smart meeting room scheduling and management system with utilization control and ad-hoc support based on real-time occupancy detection,” in *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, Ha-Long City, Quang Ninh Province, Vietnam, 2016, pp. 186–191.

[3] S. Ahmed, M. Sharmin, and S. I. Ahamed, “A Smart Meeting Room with Pervasive Computing Technologies,” in *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN’05)*, Towson, MD, USA, 2005, pp. 366–371.

[4] Panko. R.R, “Office work,” 1964, pp. 205–236.

[5] K. Padmanabh *et al.*, “iSense: a wireless sensor network based conference room management system,” in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings - BuildSys ’09*, Berkeley, California, 2009, p. 37.

[6] M. Saravanan and A. Das, “Smart real-time meeting room,” in *2017 IEEE Region 10 Symposium (TENSYMP)*, Cochin, India, 2017, pp. 1–5.

[7] O. Mussawar and K. Al-Wahedi, “Meeting scheduling using agent based modeling and multiagent decision making,” presented at the Third International Conference on Innovative Computing Technology (INTECH), London, UK, 2013, pp. 252–257.

[8] M. Tomoyuki, K. Takahashi, T. Kawamura, and K. Sugahara, “Meeting Scheduling System Using Unpleasant Notification,” presented at the International Conference on IT Convergence and Security (ICITCS), Macao, China, 2013, pp. 1–4.

[9] Z. Yu and Y. Nakamura, “Smart meeting systems: A survey of state-of-the-art and open issues,” *ACM Comput. Surv.*, vol. 42, no. 2, pp. 1–20, Feb. 2010.

[10] A. Ronzhin, A. Ronzhin, and V. Budkov, “Audiovisual speaker localization in medium smart meeting room,” presented at the 8th International Conference on Information, Communications & Signal Processing, Singapore, Singapore, 2011.

[11] H. Chen, T. Finin, Anupam Joshi, L. Kagal, F. Perich, and Dipanjan Chakraborty, “Intelligent agents meet the semantic Web in smart spaces,” *IEEE Internet Comput.*, vol. 8, no. 6, pp. 69–79, Nov. 2004.

[12] “GoGet | Easy-to-Use Meeting Room Display Solutions.” [Online]. Available: https://www.gogetcorp.com/. [Accessed: 21-May-2019].

[13] “Fishbowl | Meeting Room Display for Android Tablets and iPads.” [Online]. Available: https://fishbowlmeetings.com/. [Accessed: 21-May-2019].

[14] A. Duvnjak, “Meeting Room Booking System,” *Joan*. [Online]. Available: https://getjoan.com/. [Accessed: 21-May-2019].

[15] “Microservices.” [Online]. Available: https://martinfowler.com/articles/microservices.html. [Accessed: 21-May-2019].

[16] “Meeting Room Booking System.” [Online]. Available: https://mrbs.sourceforge.io/. [Accessed: 21-May-2019].

[17] “GPIO (General Purpose Input/Output) Definition.” [Online]. Available: https://techterms.com/definition/gpio. [Accessed: 21-Feb-2019].

[18] “PIR Motion Sensor Tutorial,” *Instructables*. [Online]. Available: https://www.instructables.com/id/PIR-Motion-Sensor-Tutorial/. [Accessed: 21-Feb-2019].

[19] “ARDUINO UNO.” [Online]. Available: http://arduino.cl/arduino-uno/. [Accessed: 21-Feb-2019].

[20] “AWS IoT Button,” *Amazon Web Services, Inc.* [Online]. Available: https://aws.amazon.com/es/iotbutton/. [Accessed: 21-Feb-2019].

[21] B. Laugwitz, T. Held, and M. Schrepp, “Construction and Evaluation of a User Experience Questionnaire,” presented at the HCI and Usability for Education and Work, Berlin, Heidelberg, 2008, pp. 63–76.