Smart Chair: An Internet of Things Case Study for a Capstone Research Project

Jing (Selena) He, Amir Atabekov, Edward Mwangi, and Donald Toler
Department of Computer Science, Kennesaw State University, Kennesaw, GA, USA
jhe4@kennesaw.edu; {aatabeko, emwangi, dtoler1}@students.kennesaw.edu

Abstract— *This paper describes the experience of a team of two undergraduate students and one graduate student to design and implement an Internet of Things (IoT)-based research project in a final year capstone project class at Kennesaw State University. The Internet of Things involves connecting physical objects to the Internet, which provide opportunity to build intelligent systems and applications by leveraging Radio-Frequency Identification (RFID), Near Field Communication (NFC), Wireless Sensor Network (WSN), and universal mobile accessibility advanced technologies. In this case study, the office/lab chairs are proposed to connect to the internet. During the one semester's capstone project class, the proof-of-concept implementation (*i.e.*, connecting one chair to the internet) is completed. To be specific, an assembled Ardunio system is responsible for scanning user ID and obtaining chair occupancy status and then sending that information to the cloud server. The collected and stored data on the cloud can be retrieved and displayed on an Android application.*

Nowadays, the undergraduate degree program has gradually evolved to include research projects in the capstone project class. Moreover, there is a growing need in our local industry for graduates with research skills in advanced technologies. This Internet of Things case study shows a good example of learning advanced technologies during the research process so that students can keep pace with the rapid rate of change in computer science.

Keywords: Internet of Things, Capstone Projects, Agile Software Development, Ardunio Boards, Cloud Services, Mobile Applications.

1. Introduction

Capstone projects are culminating experiences situated in the final year of an undergraduate students college curriculum and are designed around students demonstrating mastery of both content and application of relevant subject matter. In computer science, project-based learning model is often adopted in Capstone projects, which focus on developing case-specific problem understanding to create feasible solution options [1], [2]. In a project-based learning model, professional project-managements approach provides steps and tools to structure and support students' work. Nowadays, a growing proportion of such capstone projects undertaken at Kennesaw State University (KSU) have had a research focus, since KSU changed its status to *comprehensive university* in August 2013. Except for facilitators, the role of instructors extends to include proactive guidance of students in the challenging process of constructing not only new understanding but also developing feasible solutions in capstone research projects. This case study's smart chair system describes one such research project which connects office/lab chairs to the internet. The data of who occupancy the chair, and occupancy time duration will be collected and stored in the cloud server. The stored data can be accessed by authorized users at anytime anywhere. Moreover, the analyzed data can be used in various commercial/educational systems such as resource management, tutor time tracking management, students attendance checking, dynamic ticketing, and so on. Because of the time limitation, agile software development methodology [3] are adopted in this capstone research project. Agile methodology is described as iterative and incremental "inspect-and-adapt" approach, which provides opportunities to assess the direction throughout the development lifecycle. Every aspect of development, such as requirements, specification, design, etc. is continually revisited, so that agile methodology greatly reduces development costs and time [4].

The effective management of classrooms, halls, offices, and public spaces in any organization and the efficient and effective deployment of an organization's resources when they are needed are challenging problems. How to automatically record the activities undertaken and monitor resources usage inside rooms in real-time is usually intricate. The smart chair system described in the paper tries to solve the problem by applying Internet of Things techniques [5], [6], [7], [8], [9], [10], [11] to create an intelligent environments or decision making environments [12], [13] to facilitate the use and management aspects of our daily life. Taking students' attendance monitoring systems as a motivational example, in the past, instructors had to call the names of students or the students had to sign their names on the attendance check sheets. The former is a time consuming process, while the latter is unreliable and the attendance sheets could be lost or damaged. Subsequently, facial and voice recognition methods are proposed to verify the identity of students. Recently, Radio-Frequency (RF) communication based methods [14], [15] and smartphone based methods [16], [17] are extensively investigated. The former has some limitations to cover the whole area of a classroom by applying Near Field Communication (NFC) or Bluetooth technologies; while the latter requires the students click on smartphones. Different from the aforementioned techniques, the smart chair system can automatically check whether the chair is occupied or not by using remote sensing technologies. Moreover,the identification of the students can be automatically collected by using RFID technologies. Additionally, the lateness or leave-early information can be analyzed by integrating timestamps. Furthermore, the smart chair system has brilliant commercial prospects, which are helpful to build intelligent resource management systems, intelligent dynamic ticketing systems, etc.

Through successfully implementing the smart chair proof-of-concept research project, the students got familiar with the process of conducting research, which includes developing a research project; conducting a literature review; investigating appropriate methods and tools to implement the research project; making oral presentation of the application of the research project; and writing technique manuscript of the research findings. Moreover, this research project provides students with the opportunity to apply the knowledge and skills acquired in their courses to a specific practical problem; extend their academic experience into working with new ideas and learning new advanced technologies; demonstrate their proficiency in written and oral communication skills; extend and refine their knowledge and skill in the realization of their personal and professional goals.

The rest of this paper is organized as follows: In Section 2, the system architecture is specified. In Section 3, we describe the actual design and implementation of the smart chair system. Finally, the paper is concluded in Section 4 as well as the future work directions.

2. System Architecture

The smart chair system is based on a network of connected sensors embedded on the physical chairs to collect information, which is governed by the functionality of Internet of Things. All the collected information are uploaded to the cloud server so that any application can take advantage of data anywhere anytime when necessary. In this capstone project, we implemented the proof-of-concept experiment on a single chair. Later, we will extend it to a connected smart chair system. This project has three major components which include:

- The "thing": the chair ID and its occupancy state are stored in the cloud via Ardunio Yun [18].
- The cloud server: ThinkSpeak [19].
- The mobile application: the chair occupancy state and the user information can be displayed and monitored through mobile application.

The overall architecture of the smart chair system is described in Fig. 1. As shown in Fig. 1, RFID reader can automatically read the user's identification (denoted by UID) and passes it along to the Arduino ATmega32u4 microcontroller. The custom program checks if the pressure was applied to the pressure resistor. If the pressure applied is more than twenty pounds, the program then passes the timestamp to the Atheros AR9331 microcontroller through the bridge library [20]. Finally the bridge library issues a HTTP POST request to the ThingSpeak API [19] along with the UID of the scanned RFID card.

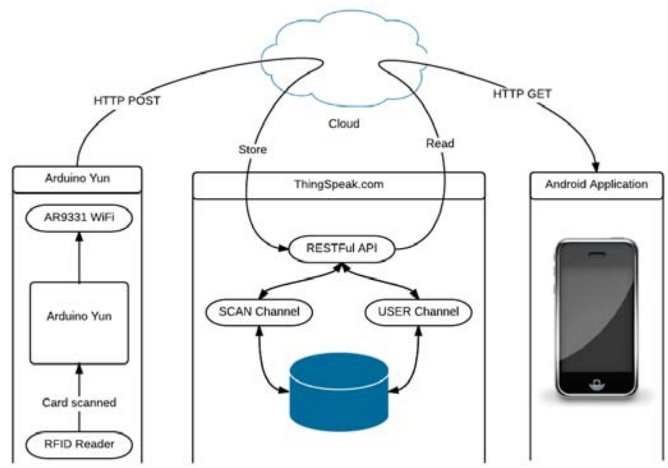


Fig. 1: Smart Chair System Architecture.

Moreover, the Android application is also developed to let administrators or other interested users monitor when or who has scanned his card and sat on the chair. The overview of the Android architecture is also illustrated in Fig. 1. The Android application issues an HTTP GET request to ThingSpeak's JSON feed. Each HTTP GET request is authenticated against an API key (more details will be mentioned in the next section). When submitting a HTTP GET request, the API key must be included in the URL string. Finally, the retrieved data will be displayed on the Android application.

In more detail, the data flow of the whole system is summarized in Fig. 2. First RFID scan is performed and the card UID is read and transferred to the Arduino microcontroller, then the program on Arduino transfers the data to the AR9331 chip with bridge library. AR9331 being a WiFi chip submits a HTTP POST request to the RESTFul API of the ThingSpeak.com. Likewise in order to display the latest scans, Android application submits HTTP GET request to the RESTFul API and fetches the data in JSON format. Finally Android application parses the JSON and displays the latest scans.

3. System Design and Implementation

In this section, we will mention the details to design and implement the smart chair system. Before we move to the

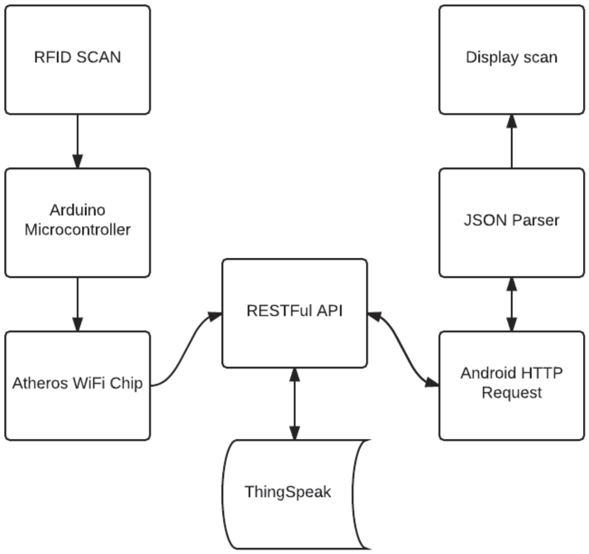


Fig. 2: System Data Flow Diagram.

technical details, the system requirements are summarized and listed in Table 1.

Table 1: Smart Chair System Requirements

Requirement	Description
Read RFID tag UID with reader	Write Arduino program which will read the UID off the RFID tag when in close proximity
Sense when pressure is applied	Connect pressure resistor and write corresponding program which detects when pressure is applied
Program indication lights	Implement LED indication lights in the program to know the system status
Find and configure IoT restful service to store data	ThingSpeak chosen as backend for the project. Corresponding USER and SCAN channels will be configured to store data.
Send the UID data to ThingSpeak	Write Arduino program which will send the UID data to ThingSpeak channel. Yun Bridge library is used for HTTP POST requests.
Implement Login screen	Login screen will be implemented on Android application to authenticate users
JSON Parser	A JSON parser will be written to parse the JSON data returned from ThingSpeak channels.
Develop Android Client	Android application will be developed to view the latest scans performed.

Next, we are ready to illustrate the system implementation details.

3.1 Hardware

Fig. 4 shows all the hardware components which are assembled with Ardunio Yun (shown in Figure 3):

- RFID Reader: SainSmart RC522 RFID reader was chosen for reading UID off user's RFID cards. RC522 is easy to connect to Arduino, since a library [21] was already written to communicate with the reader through Ardunio Serial Peripheral Interface (SPI) bus [22].
- Pressure Sensor: Interlink 402 pressure resistor was chosen to detect if a person is currently sitting on a chair.
- Miscellaneous: Three LEDs (Red, Green and Blue) were used to indicate the status of data transmission or the success of card read. Three 180 ohm resistors were used for LEDs and one $10k$ ohm resistor was used for pressure sensor.

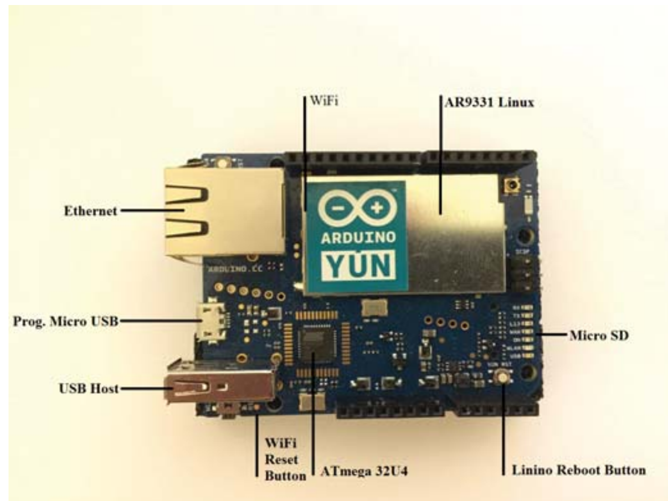


Fig. 3: Ardunio Yun Board.

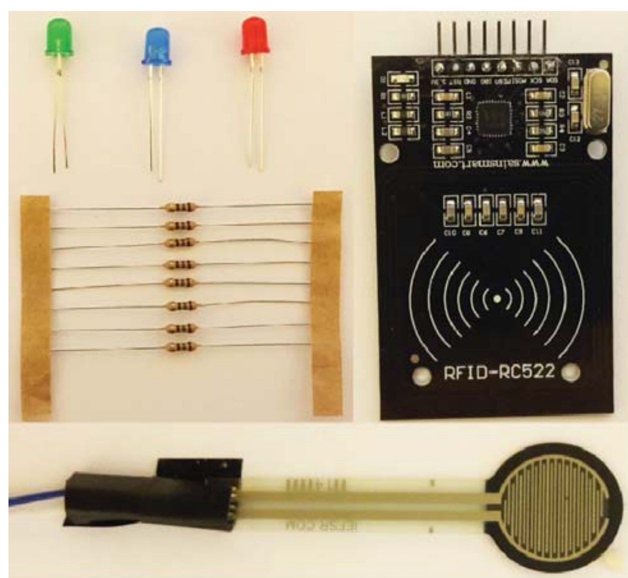
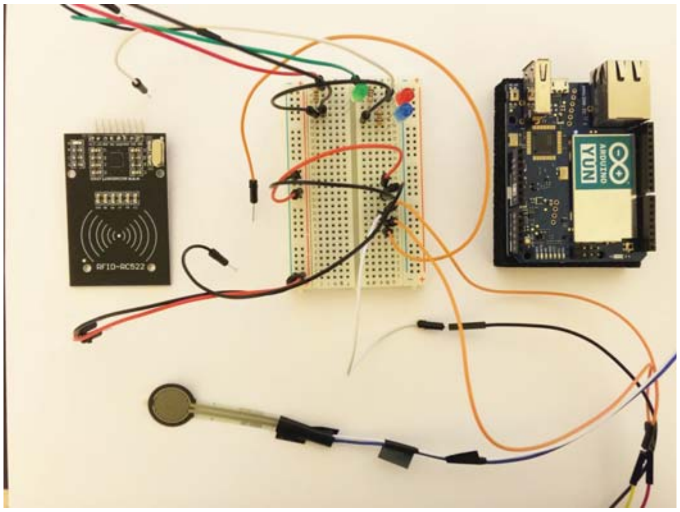
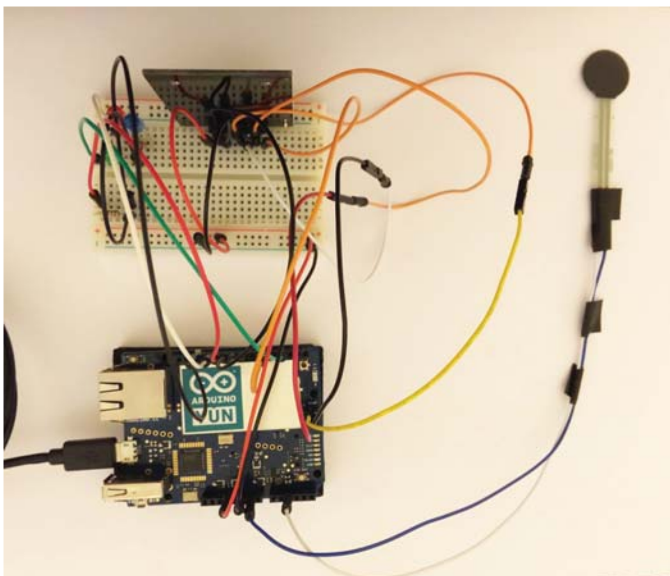


Fig. 4: Hardware Components of Smart Chair System.

The unassembled system and fully assembled system are shown in Fig. 5 (a), and (b) respectively.



(a)

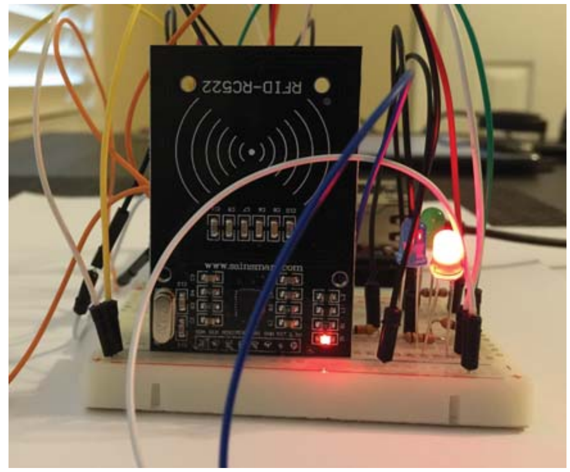


(b)

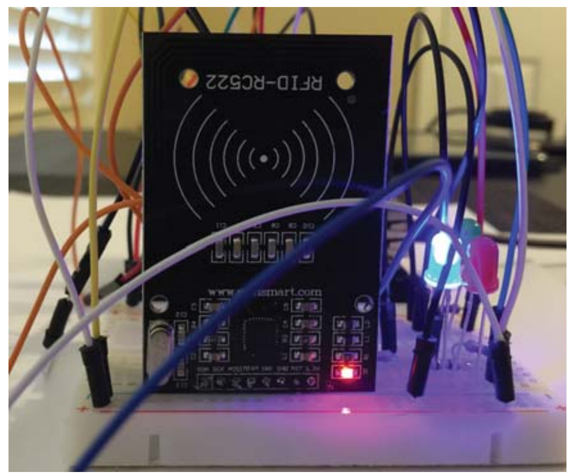
Fig. 5: (a) Unassembled System; (b) Assembled System.

3.2 Reading and Processing Data

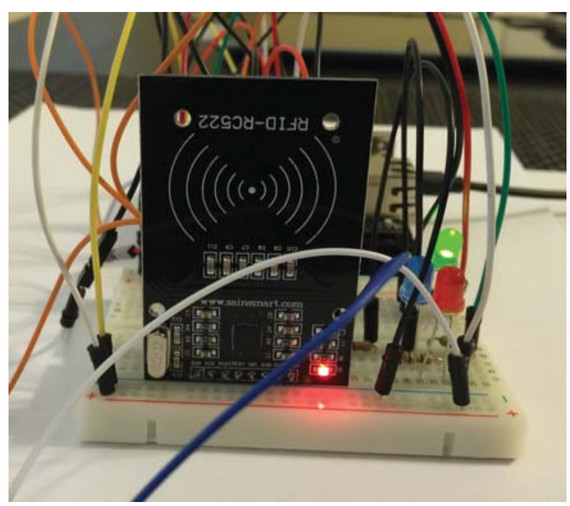
The assembled smart chair system has three LEDs, which indicate success or failure of data transmission. As we already mentioned, the system will read UID through RFID reader and the pressure through pressure sensor. If the sensed pressure is greater than twenty pounds, the corresponding timestamp will be collected. As shown in Fig. 6 (a), the red colored LED indicates that the scan was unsuccessful or not enough pressure was applied to the pressure sensor. Fig. 6 (b) shows the blue colored LED is turned on, which indicates that enough pressure was applied and UID was successfully read. Finally the green colored LED indicates that data transmission to the ThingSpeak was successful as displayed in Fig. 6 (c).



(a)



(b)



(c)

Fig. 6: LED Indications in Smart Chair System: (a) RED indication; (b) BLUE indication; (c) GREEN indication.

3.3 Data in the cloud

Whenever there is a successful scan with enough pressure, the UID and timestamp data will be sent to ThingSpeak. ThingSpeak uses a concept of "channels", where each channel can be configured with different labels to hold different kinds of data. Our system is configured with two channels. The first channel named "USER" holds user information, while the second channel named "SCAN" holds the data related to RFID card scan. For "USER" channel we have labeled eight fields indicating card key, name, gender, email, phone number, major, address and faculty status of a user as shown in Fig. 7.

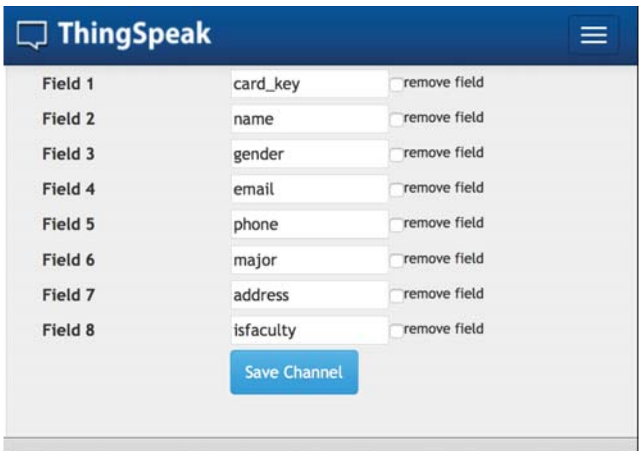


Fig. 7: USER Channel Creation Screen on ThingSpeak.

The "SCAN" channel holds only one label named "UID" along with the default label indicating timestamp. The reason is that we don't need to hold any extra information, which can easily be found in "USER" channel with the help of card UID. ThingSpeak provides a data feed for each channel in JSON format, which can be retrieved by any client application such as by Android, or iOS mobile applications. Fig. 8 illustrates the JSON feed for "USER" channel. The first object in JSON feed is called *channel*. It contains information about the channel such as the "id" of the channel, name of the channel, names of all fields in the channel and the "id" of the last entry.

Likewise the JSON field for "SCAN" channels is illustrated in Fig. 9. This channel has only one field name "UID" which holds the card UID. In addition to UID, time stamp field is provided by default. There's no need to store any more information other than UID, since the data related to particular user can be located in "USER" channel.

The first object in "SCAN" channel as always, contains information about the fields of the channel, its name and id. The second object which is a JSON array named "feeds" contains the actual feeds of the channel.


```
{
    "channel": {
        "id": 27566,
        "name": "USER",
        "field1": "card_key",
        "field2": "name",
        "field3": "gender",
        "field4": "email",
        "field5": "phone",
        "field6": "major",
        "field7": "address",
        "field8": "isfaculty",
        "created_at": "2015-02-23T01:31:52Z",
        "updated_at": "2015-03-17T14:12:14Z",
        "last_entry_id": 2
    },
    "feeds": [
        {
            "created_at": "2015-03-17T14:11:15Z",
            "entry_id": 1,
            "field1": "E3 CC 50 CE",
            "field2": "Amir Atabekov",
            "field3": "Male",
            "field4": "sample@gmail.com",
            "field5": "222-777-2187",
            "field6": "Computer Science",
            "field7": "1501 Miramar St, Valdosta, GA, 31602",
            "field8": "No"
        },
        {
            "created_at": "2015-03-17T14:12:14Z",
            "entry_id": 2,
            "field1": "94 CB 59 CB",
            "field2": "Selena He",
            "field3": "Female",
            "field4": "sample@gmail.com",
            "field5": "222-777-0000",
            "field6": "Computer Science",
            "field7": "100 Chastain Rd, Kennesaw, GA, 30102",
            "field8": "Yes"
        }
    ]
}
```


Fig. 8: JSON Feed for USER Channel.


```
{
    "channel": {
        "id": 27574,
        "name": "SCAN",
        "field1": "UID",
        "created_at": "2015-02-23T04:24:53Z",
        "updated_at": "2015-03-10T20:02:48Z",
        "last_entry_id": 215
    },
    "feeds": [
        {
            "created_at": "2015-02-24T02:07:02Z",
            "entry_id": 116,
            "field1": "E3 CC 50 CE"
        },
        {
            "created_at": "2015-02-24T02:07:17Z",
            "entry_id": 117,
            "field1": "E3 CC 50 CE"
        },
        {
            "created_at": "2015-02-24T02:07:32Z",
            "entry_id": 118,
            "field1": "E3 CC 50 CE"
        },
        {
            "created_at": "2015-02-24T02:24:16Z",
            "entry_id": 119,
            "field1": "E3 CC 50 CE"
        },
        {
            "created_at": "2015-02-24T04:09:59Z",
            "entry_id": 120,
            "field1": "E3 CC 50 CE"
        },
        {
            "created_at": "2015-02-24T04:15:39Z",
            "entry_id": 121,
            "field1": "E3 CC 50 CE"
        },
```


Fig. 9: JSON Feed for SCAN Channel.

3.4 Retrieving the data

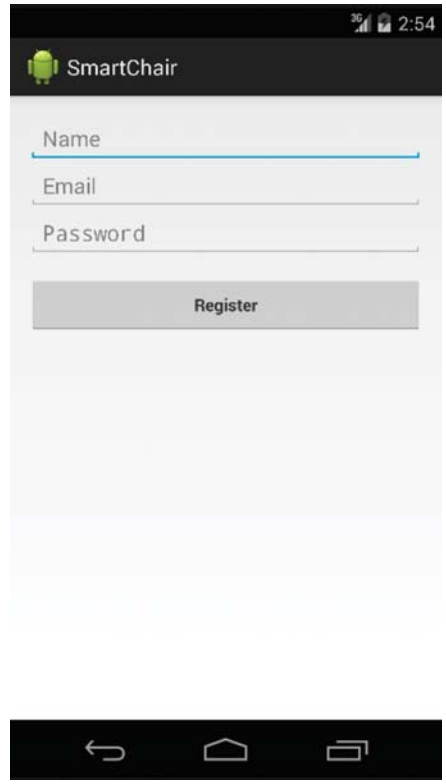
In this project, we also implement Android Application to retrieve data stored in the cloud server. As we mentioned before, the Android application needs to issue HTTP GET requests to ThingSpeak's JSON feed. Each HTTP GET request is authenticated against an API key which is generated when a new channel is configured (shown in Fig. 8 and 9). When submitting a HTTP GET request, the API key must be included in the URL string. Since ThingSpeak requires API keys to be included in every request, it is helpful to implement some form of authentication on Android application as well. Fig. 10 (a), and (b) show the registration and login screens of the Android application respectively.

Once the "Sign in" button is pressed, an Android AsyncTask is executed in the background to first fetch the results of the "USER" channel. Once the "USER" channel feed is obtained, it is stored in a HashMap data structure locally. Then a second HTTP GET request is sent to fetch the results of the "SCAN" channel. Once obtained, the data from the "SCAN" channel is merged with data we stored earlier for "USER" channel. Finally the merged records are displaced as illustrated in Fig. 11. As shown, the name of the user is displayed followed by the timestamp when the last card scan was performed. The yellow progress bar indicates the loading progress of retrieving the data. If it is full, it means that all the data was successfully fetched and stored locally.

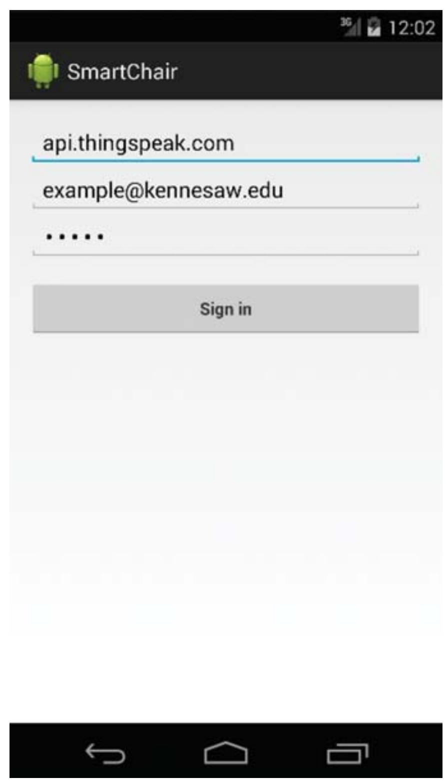
4. Conclusion and Future Work

The proof-of-concept experiment of the smart chair system has been completely implemented in this capstone research project. An Ardunio board based system is assembled and implemented to get user ID through RFID reader and chair occupancy status by pressure sensor. All data is uploaded to the ThingSpeak cloud. Moreover, an Android application to view the stored data has also been implemented. Through successfully implementing the project, the whole team gained invaluable knowledge in hardware design, hardware prototyping, cloud services, and android application development. After developing this project, we can recommend Ardunio hardware to those interested in implementing Internet of Things systems. Since, Ardunio hardware is easy to use and can be intergraded with numerous different modules. Moreover, the learning curve for programming on the devices is not sharp as the Ardunio developer community is large and there are many tutorials available on the Internet.

It should be noted that the possibilities exist not only in connecting the chair to internet but also in the use of real time data stored in the cloud to create new applications. This smart chair system not only offers the solution to a problem but also lays the foundation for series of future projects (such as intelligent parking system, dynamic ticketing system, *etc.*) Because of the time limitation, only the proof-of-concept



(a)



(b)

Fig. 10: (a) Registration Screen; (b) Login Screen of Mobile App.

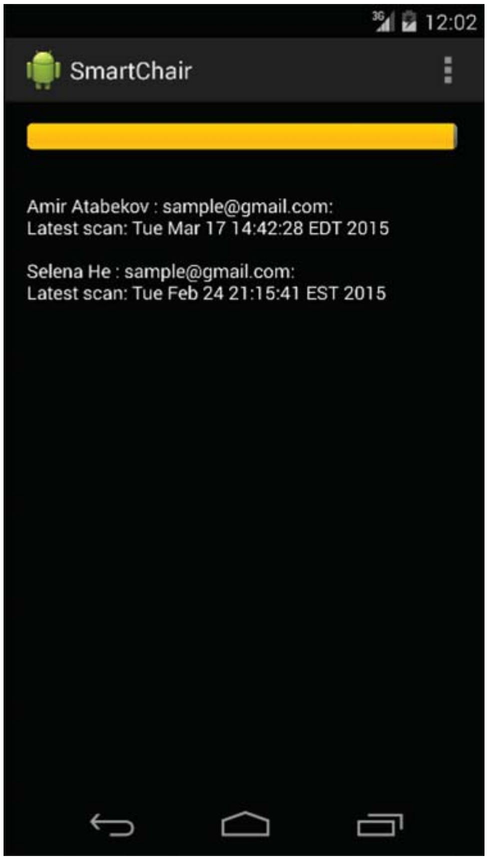


Fig. 11: Data Display Screen of Mobile App.

experiments are completed in the capstone project. Lots of additional improvements to the system can be implemented. So far only one chair connects to the internet. It is more practical to connect all chairs in a room to the internet. Hence, constructing a wireless sensor network so that every chair in a room can be connected to the internet is one research direction. Moreover, a simple mobile application is implemented in this project. Advanced application can be designed and developed, such as to show the chair map on the screen. Additionally, the occupancy status of each chair can be graphically displayed/viewed on the screen. More advanced functionality could be to let users can interact with the graph to reserve a chair. Finally, the collectted/analyzed data could be used for additional purposes, such as resource management, attendance checking, or tutor time tracking management.

Acknowledgment

This work is funded in part by the Kennesaw State University's Office of the Vice President of Research (OVPR) Pilot/Seed Grant, by the College of Science and Mathematics Interdisciplinary Research Opportunities (IDROP) Program, and by the Department of Computer Science Mini-Research grant.

References

[1] P.C. Blumenfeld, E. Soloway, R.W. Marx, J.S. Krajcik, M. Guzdial, and A. Palincsar, *Motivating project-based learning: Sustaining the doing, supporting the learning*, Educational Psychologist, Vol. 26, No. 3, pp. 369-398, 1991.

[2] H. Roessingh, and W. Chambers, *Project-based learning and pedagogy in teacher preparation: staking out the theoretical mid-ground*, International Journal of Teaching and Learning in Higher Education, Vol. 23, No. 1, pp. 60-71, 2011.

[3] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*, Prentice Hall PTR Upper Saddle River, NJ, USA, 2003.

[4] S. Dorairaj, J. Noble, and P. Malik. *Knowledge Management in Distributed Agile Software Development*, Agile Conference (AGILE), pp. 64-73, Dallas, TX, Aug. 2012.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, *Internet of Things (IoT): A vision, architectural elements, and future directions*, Future Generation Computing System, 29, pp. 1645 Ű 1660, 2013.

[6] K. Ashton, *That "Internet of Things" Thing*, RFiD Journal, Vol. 22, pp. 97 Ű 114, 2009.

[7] X. Jia, O. Feng, T. Fan, and Q. Lei, *RFID technology and its applications in internet of things (IoT)*, 2nd IEEE conference on Consumer Electronics, Communications and Networks (CECNet'12), pp. 1282 - 1285, 2012.

[8] C. Sun, *Application of RFID technology for logistics on internet of things*, AASRI Conference on Computational Intelligence and Bioinformatics, pp. 106 - 111, 2012.

[9] J. He, S. Ji, Y. Pan, and Y. Li, *Constructing Load-Balanced Data Aggregation Trees in Probabilistic Wireless Sensor Networks*, IEEE Transactions on Parallel and Distributed Systems (TPDS), Vol. 25, No. 7, pp. 1681 - 1690, July, 2014.

[10] J. He , S. Ji, R. Beyah, Y. Xie, and Y. Li, *Constructing Load-Balanced Virtual Backbones in Probabilistic Wireless Sensor Networks via Multi-Objective Genetic Algorithm*, Transactions on Emerging Telecommunications Technologies (ETT), Vol. 26, No. 2, pp. 147 - 163, February, 2015.

[11] J. He, S. Ji, R. Beyah, and Z. Cai, *Minimum-sized Influential Node Set Selection for Social Networks under the Independent Cascade Model*, ACM MOBIHOC 2014, pp. 93 - 102, 2014.

[12] C. Ramos, G. Marreiros, R. Santos, and C.F. Freitas, *Smart Offices and Intelligent Decision Rooms*, pp. 851-880, Handbook of Ambient Intelligence and Smart Environments, Spriner, New York, NY, USA, 2010.

[13] J. Reijula, M. Grohn, M. Muller, and K. Reijula, *Human well-being and flowing work in an intelligent work environment*, Intelligent Buildings International, Vol. 3, Issue 4, pp. 223-237, 2011.

[14] H. Chen, *Intelligent Classroom Attendance Checking System Based on RFID and GSM*, Advanced Materials Research, Vol. 989-994, pp. 5532-5535, 2014.

[15] M. Zhi, and M. M. Singh, *RFID-Enabled Smart Attendance Management System*, Future Information Technology, Vol. 329, pp. 213-231, 2015.

[16] M. Choi, J.-H. Park, and G. Yi, *Attendance Check System and Implementation for Wi-Fi Networks Supporting Unlimited Number of Concurrent Connections*, International Journal of Distributed Sensor Networks, pp. 1 - 10, 2014.

[17] A. A. Kumbhar, K. S. Wanjara, D. H. Trivedi, A. U. Khairatkar, and D. Sharma, *Automated Attendance Monitoring System using Android Platform*, International Journal of Current Engineering and Technology, Vol. 4, No. 2, pp. 1096-1099, 2014.

[18] Ardunio Yun Board, Available online: http://arduino.cc/en/Main/ArduinoBoardYun (accessed in March 2015).

[19] ThingSpeak, Available online: https://thingspeak.com/ (accessed in March 2015).

[20] Anduino Bridge Library, Available online: http://arduino.cc/en/Reference/YunBridgeLibrary (accessed in March 2015).

[21] Arduino RFID Library for MFRC522, Available online: https://github.com/miguelbalboa/rfid (accessed in March 2015).

[22] Ardunio Serial Peripheral Interface (SPI), Available online: http://arduino.cc/en/Reference/SPI (accessed in March 2015).