Proceedings of the Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020)
IEEE Xplore Part Number:CFP20K25-ART; ISBN:978-1-7281-4889-2

Smart Restaurant System using RFID Technology

Jedidiah Harpanahalli*, Kevin Bhingradia†, Pranav Jain‡, Jayasudha Koti§
Department of Electronics and Telecommunication
Saint Francis Institute of Technology, Mumbai, India
Email: chinnikanna.h@outlook.com*, kevinbhingradia@gmail.com†, prnvjn@gmail.com‡, jayasudhakoti@sfit.ac.in§

Abstract—With a move towards a digital India, digitization has to be ensured in all aspects of the society. In this paper a RFID based restaurant management system by using the concepts of open source technologies like Python and Raspberry Pi is presented. This system is introduced as a solution to the bottleneck caused by the cashiers. Focusing on selfservice, the proposed system aims to develop a digital, contactless and secure restaurant environment which will enable patrons to seamlessly *Select, Scan and Eat* their desired food items. Identification of food items along with the payment is done using Radio Frequency Identification (RFID) technology wherein each food item is marked using adhesive RFID tags, the amount for the same is deducted automatically from the patron's digital wallet. This results in significant advancement towards automatic management and reduction in manual labour in a restaurant environment.

Index Terms—RFID, Digital wallet, Adhesive RFID tags

I. INTRODUCTION

In conventional canteen and restaurant systems people used to and still wait in long queues for food selection and payment. This leads to chaos and confusion in canteens and restaurants during the peak hours in order to overcome this we propose a *smart restaurant* system. This avoids confusion and speeds-up the entire process. In this century, people live a life which is solely dependent on technology. New innovations are made to make out life less demanding, calm and more agreeable. The primary goal of advancement has been to extend capability and diminishing efforts [1]. This model puts forward a method to streamline self-service restaurant management, the system consists of two levels, the upper level being controlled by the Raspberry Pi and the lower by the Arduino. The Raspberry Pi acts as a master which not only controls the backend processing and payment but also issues commands to the Arduino. The Arduino on the other hand executes the commands issued by the Raspberry Pi by collecting information from the RFID readers. The entire system will be operated using a GUI running on the Raspberry Pi which would display the scanned items, the final cost and the option to proceed with the payment. The rest of the paper is described as follows Section II discusses about the overview of the system, Section III discusses about the hardware implementation, Section IV discusses about the experimental results and finally Section V concludes the paper.

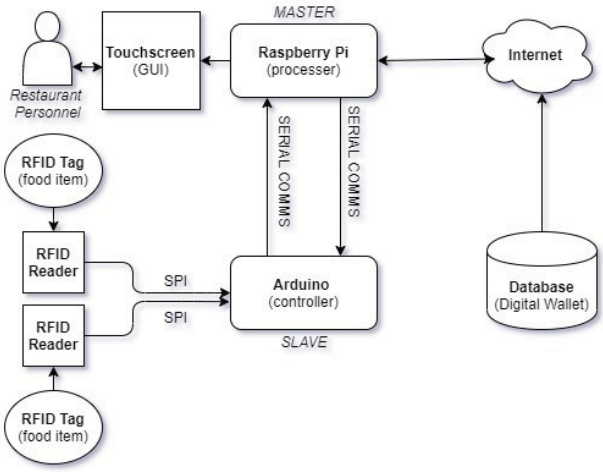


Fig. 1: Block Diagram

II. OVERVIEW OF THE SYSTEM

A. Block Diagram

The model comprises of two sub-systems as highlighted in Fig.1, one system being the master and the other one being the slave. The master system consists of the Raspberry Pi which handles the processing, GUI and the transaction. The calculated payment amount will be debited from the users digital wallet. The interfacing with the slave system (Arduino) is done using serial communication. The slave system consists of an Arduino, which in turn is connected to multiple RFID readers, the communication between the readers and the Arduino takes place by using Serial Peripheral Interface (SPI). Each RFID reader is used to identify 13.56Mhz passive RFID tags which would be used to identify the food items.

B. Work Flow

The Fig. 2 highlights the master Sub-System's working. On initialization, the subsystem will firstly check its connection with the Arduino and the database, if the connection is successful the GUI boots up. Once the boot up is complete the system will wait until a button is pressed. The scan button when pressed, issues a command from the Raspberry Pi to the Arduino requesting the scanned data. The data received is verified, displayed on the GUI and the total due amount is calculated. If the payment is authorized by pressing the pay button, the calculated amount is debited from the user's linked digital wallet. The Fig. 3 highlights the slave sub-system's working. Once the system boots-up the connection with the RFID readers is initialized, this is followed by scanning of

RFID tags (if any present), the scanned tag IDs are then sent to the Raspberry Pi for further processing.

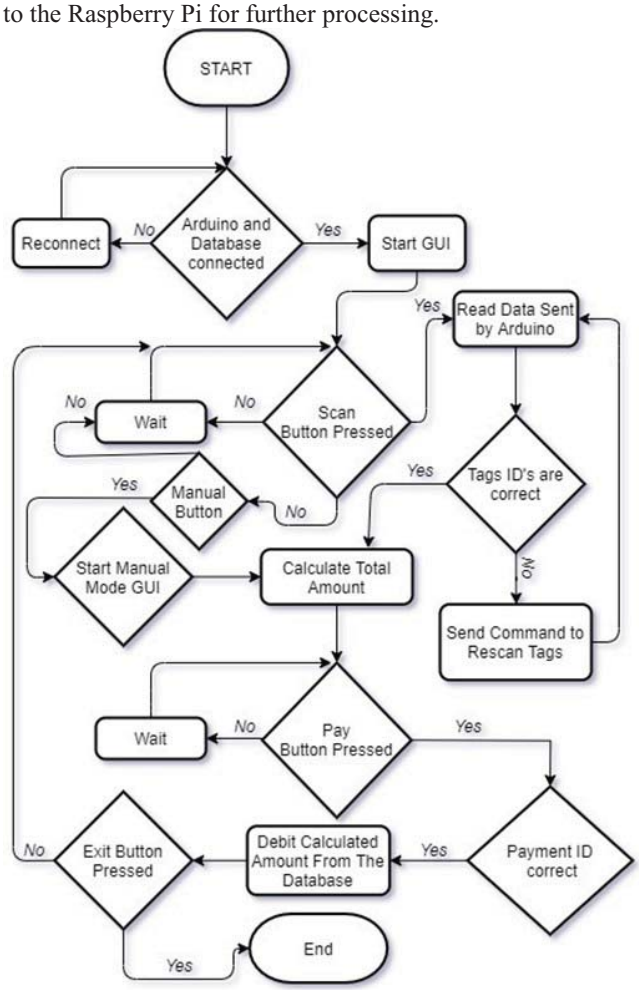


Fig. 2: Master Sub-System

C. Circuit Diagram

Fig. 4 highlights the circuit diagram of the entire model. The slave subsystem that consists of the Arduino and the RFID readers communicate with each other using the *SPI* protocol. The *SPI* protocol uses, *Master-In-Slave-Out(MISO)*, *Master-Out-Slave-In(MOSI)*, *Reset(RST)*, *Serial-Clock(SCK)* and *Slave-Select(SS)* pins for communicating. In the slave subsystem the Arduino acts as the master. *MISO* pin is used by the Arduino to send commands to the RFID reader, the reader in response to the *MISO* signal sends data back to the Arduino by using the *MOSI* pin. *SCK* pin is used by the Arduino to send clock signals to the readers, this enables synchronization between the both. *SS* pin is used to toggle between the different RFID readers. *RST* this pin is used to reset connection between the readers and the Arduino. The entire reader array works on 3.3v that is supplied by the Arduino. The data that is collected from the readers is sent to the Raspberry from the Arduino Via a *serial USB* connection. The Raspberry Pi is connected to a touchscreen using a *High-Definition Multimedia Interface (HDMI)* connection.

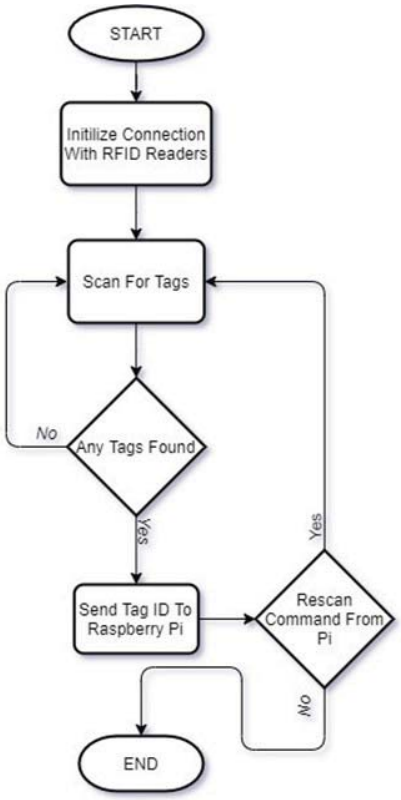


Fig. 3: Slave Sub-System

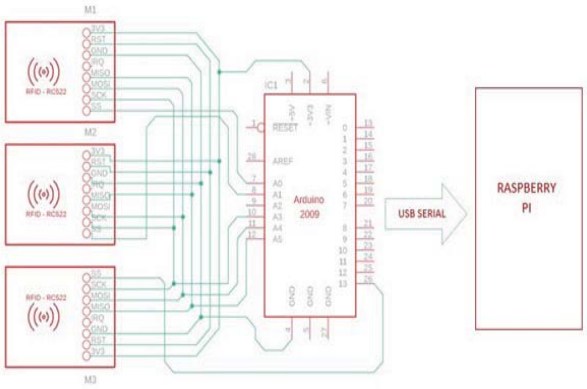


Fig. 4: Slave Sub-System

III. Design and Implementation

A. Design of Scanning System

Fig. 5 highlights the top view of the model's scanning surface. The top scanning surface is covered by a thin glass layer, below which RFID readers are embedded to scan the RFID tags placed over the surface.

This acts as a smooth surface over which the food tray can easily be slid. Fig. 6 represents the restaurant personnel's side, which consists of a touchscreen panel running the system's GUI. This enables the restaurant personnel to view the scanned items and authorize payments. patron's side is shown in Fig. 7, once the patron enters the restaurant he/she will be provided with a food tray, the tray will be used to place any

food items that the patron wishes to purchase, once the items are selected the patron will come to the scanning table and slides his/her food tray containing the food items along the scanning surface. The scanned food items will be displayed on the restaurant personnel's screen.

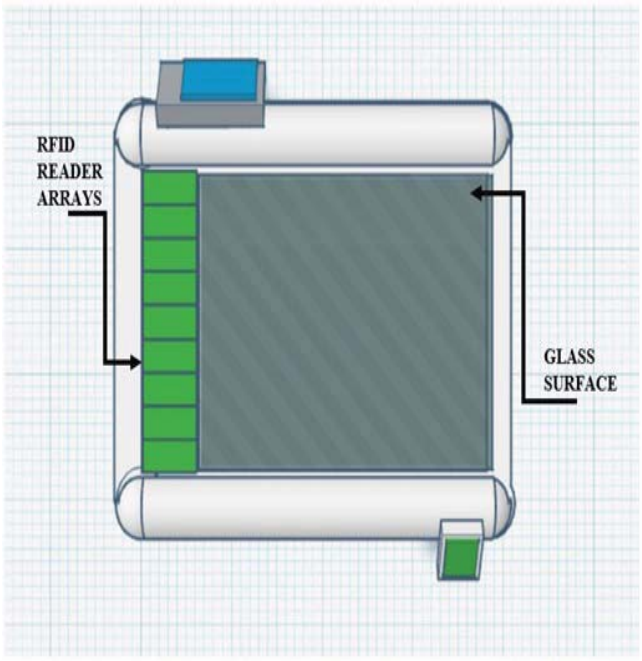


Fig. 5: Top-View

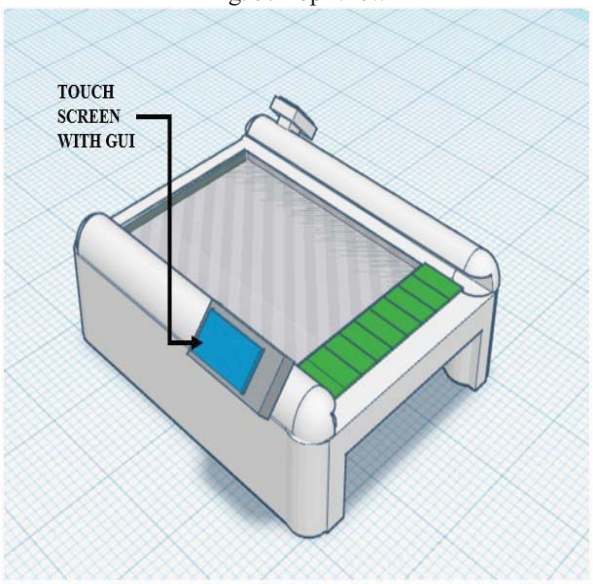


Fig. 6: Restaurant Personnel-View

If the items are scanned properly the payment will be authorized, by prompting the user to tap his/her paymentID onto the payment RFID scanner. Subsequently, the calculated total amount will be debited from the patron's linked digital wallet.

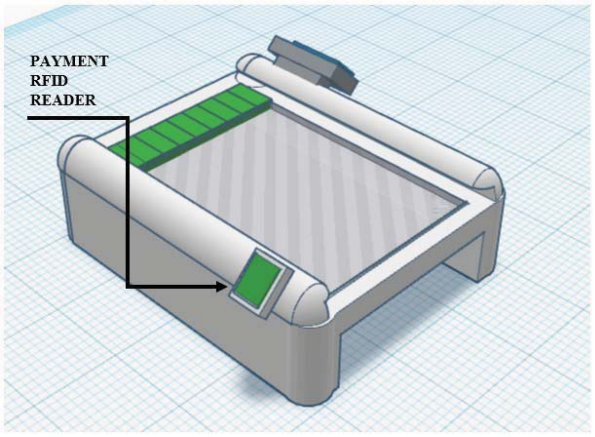


Fig. 7: patron-View

B. Passive RFID tags

RFID tag is a tiny device that stores and forwards data into a RFID reader [2]. Our model uses adhesive 13.56Mhz passive RFID tags. These tags are used to identify each food item. The food items will be placed individually in small containers, each container being marked by a RFID tag. These tags are cheap, washable, and have a long lifespan which makes them an ideal choice for marking the food items.

C. Payment Unique Identification (UID)

Each patron will be issued a RFID card that will be used for payment purpose. This will act as the paymentUID card. The card will contain the UID of the patron that will help identify the patrons digital wallet. This card will be scanned at the checkout once a payment is initialized

D. RFID Reader

MFRC522 is the RFID reader being used in our system, it uses SPI to interface with the microcontroller. It works on 13.56Mhz frequency, which is the same frequency as that of the tags. In addition to SPI protocol, this module also supports I^2C protocol.

E. Raspberry Pi

Raspberry Pi is a credit-card-sized single-board computer developed in the UK by Raspberry Pi foundation with the intention of stimulating the teaching of basic computer science in schools. The foundation provides Debian and Arch Linux ARM distributions and also Python as the main programming language, with the support for BBC BASIC, C and Perl [3].

F. Arduino Mega

Arduino Mega R3 2560 is the microcontroller used for the project. The Arduino Mega R3 2560 is a board based on the ATmega2560 microcontroller [4]. It acts as an interface between RFID readers and the Raspberry Pi.

G. Graphical User Interface

The GUI is built using the python 'Tkinter' framework. The main GUI screen is highlighted as shown in Fig. 8. Being built on python further modification and optimization is made quite simple. A manual mode as shown in Fig. 9 is also provided as a backup.

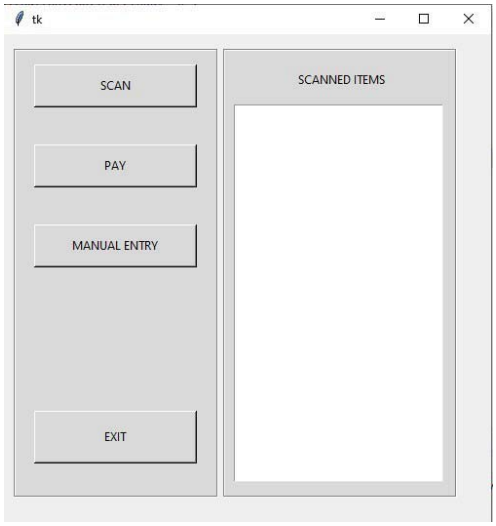


Fig. 8: Graphical User Interface

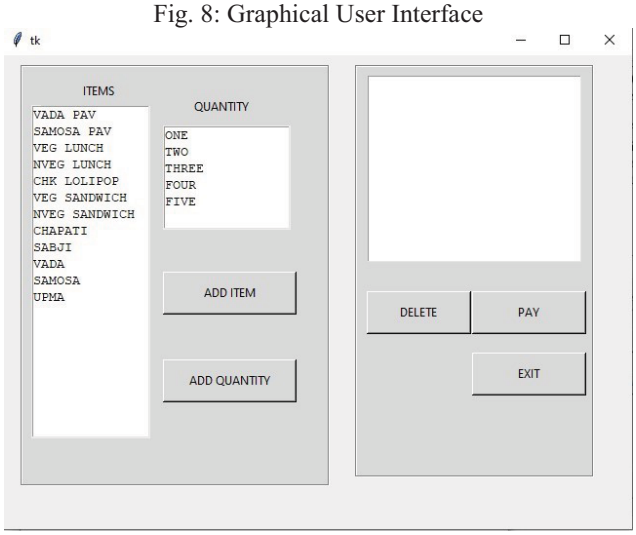


Fig. 9: Manual Mode Graphical User Interface

H. Software

Arduino IDE has been used to program the Arduino and Arduino modules. Raspbian Operating system has been loaded on the SD card with NOOBS OS which is necessary to boot the Raspberry pi. The coding of the sensors and modules attached to the raspberry pi has been done on the python platform. Coding on python can be done either by using the shell or by python software IDLE [5]

```
SELECT * FROM Accounts_Customer
```

Export	Cust_Name	Cust_Phno	Cust_UID	Cust_Balance	Cust_Accno
1	Jedidiah	993456789	432B54D3	1500	55446675334
2	Pranav	987654321	86C2061F	5000	23211889900
3	Kevin	913456789	8622651F	567	12234454666
4	Manish	921145678	8612D11F	890	234431200006

Fig. 10: Database Rows

I. Digital Wallet Database

The digital wallet enables cashless transactions within the system, and thus is one of the most crucial parts. Each wallet account is associated to a patron from which the payment amount will be debited automatically.

The database for the same is created using Structured Query Language (SQL). Fig. 10 highlights the top few rows of the digital wallet database. *Cust Name* is a column used to identify the name of the patron to whom the digital wallet belongs, *Not-Null* SQL constraint has been applied to this column to ensure that the column is never left empty. *Cust phno* contains the phone number of the patron, the column has been programmed with the *Unique* and *Not-Null* constraints to prevent any duplicate entry and additionally ensures that no column is left blank. *Cust UID* column uniquely identifies each patron, thus acts as the *primary key* for the database. This ID is obtained by scanning the patron's payment tag during the transaction phase, *Unique* and *Not-Null* constraints are applied to this column. *Cust Balance* is a numeric column that holds the balance amount of the patron's wallet, any transactions when debited would be reflected in this column. *Cust Accno* column holds the patron's account number, this column is also of the numeric type with, *Unique* and *Not-Null* constraints applied.

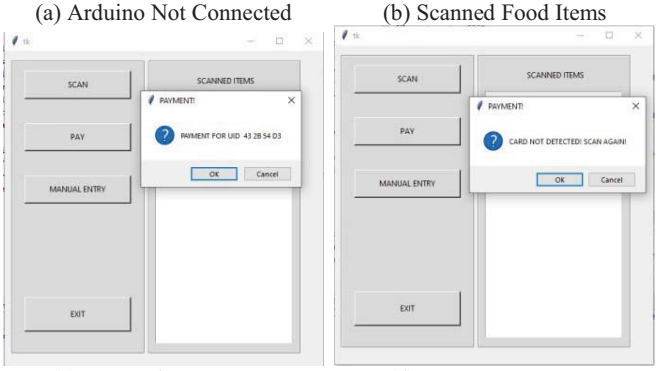
IV. Results

Once the system starts it checks its connection with the Arduino and the database, in case the Arduino is not connected an error message requesting the user to connect the Arduino as shown in Fig. 11(a) Fig. 11(b) shows the items scanned by the system, which can be used by the restaurant personnel to cross-verify the food items placed on the tray.

Fig. 11(c) shows patron's scanned payment UID which is used to identify the patron's digital wallet, if the payment UID card is not scanned properly an error message is shown on the GUI requesting the patron to rescan his/her payment UID card as shown in Fig. 11(d).

If the payment is authorized the Raspberry Pi calculates the total and proceeds with the transaction. The scanned payment UID of the patron is used to query the linked digital wallet from the database, in this case the digital wallet account with UID *432B54D3* is selected as highlighted in Fig. 12. The calculated total amount is then debited from the digital wallet, this can be seen by comparing the *Cust Balance*

(a) Arduino Not Connected (b) Scanned Food Items



(c) Scanned PaymentID (d) Rescan PaymentID

Fig. 11: Start-up and Scanning

```
SELECT * FROM Accounts_Customer
```

Export	Cust_Name	Cust_Phno	Cust_UID	Cust_Balance	Cust_Accno
1	Jedidiah	993456789	432B54D3	1500	55446675334
2	Pranav	987654321	86C2061F	5000	23211889900
3	Kevin	913456789	8622651F	567	12234454666
4	Manish	921145678	8612D11F	890	234431200006

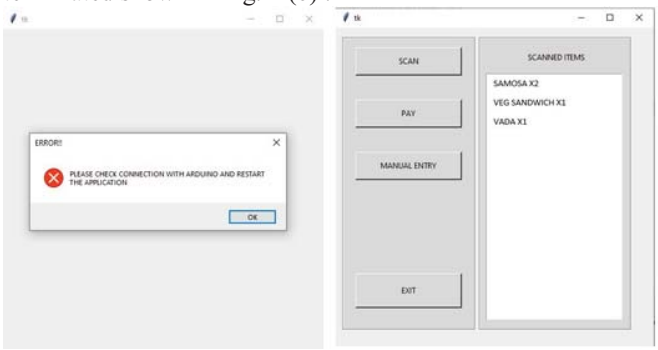
Fig. 12: Database Row Selected

```
SELECT * FROM Accounts_Customer;
```

Export	Cust_Name	Cust_Phno	Cust_UID	Cust_Balance	Cust_Accno
1	Jedidiah	993456789	432B54D3	1000	55446675334
2	Pranav	987654321	86C2061F	5000	23211889900
3	Kevin	913456789	8622651F	567	12234454666
4	Manish	921145678	8612D11F	890	234431200006

Fig. 13: Database Balance Updated

column value for UID *432B54D3*. This is followed by a *payment successful* messagebox shown in Fig.14(a), if the patron's wallet doesnt have enough funds for the payment an error message is displayed on the GUI and the transaction is terminated shown in Fig.14(b) .



(a) Payment Successful (b) Insufficient Balance

Fig. 14: Payment Success and Insufficient Balance

V. Conclusion

In order to avoid the confusion, long queues and bottlenecks caused by cashiers during peak hours, we the authors presented a novel system that performs automatic food-scanning and payment, focused on improving service quality at restaurants. The proposed smart restaurant system has been addressed and implemented using Arduino, Raspberry pi, Python, SQLite database and RFID technology. The same has been tested for different conditions like start-up and scanning process, database transactions and payment execution, the results have been discussed in the results section. It further improves the system by endorsing cashless transactions and streamlining the otherwise chaotic restaurant environment. In addition to that, it also aids to diminish human intervention in restaurant operations. Thus this system comes in handy as an admirable solution in restaurant operations.

References

[1] S. Ahmed, T. M. Tan, A. M. Mondol, Z. Alam, N. Nawal and J. Uddin," Automated Toll Collection System Based on RFID Sensor," 2019 International Carnahan Conference on Security Technology (ICCST), CHENNAI, India, 2019, pp. 1-3. doi: 10.1109/CCST.2019.8888429

[2] N. S. Kumar, B. Vuayalakshmi, R. J. Prarthana and A. Shankar, "IOT based smart garbage alert system using Arduino UNO," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 1028-1034. doi: 10.1109/TENCON.2016.7848162

[3] G. S. Sundaram et al., "Bluetooth communication using a touchscreen interface with the Raspberry Pi," 2013 Proceedings of IEEE Southeastcon, Jacksonville, FL, 2013, pp. 1-4. doi: 10.1109/SECON.2013.6567448

[4] Bhavke, A., & Pai, S. (2017, January). Advance automatic toll collection & vehicle detection during collision using RFID. In 2017 International Conference on Nascent Technologies in Engineering (ICNTE) (pp. 1-5). IEEE.

[5] S. Goyal, P. Desai and V. Swaminathan, "Multi-Level Security Embedded With Surveillance System," in IEEE Sensors Journal, vol. 17, no. 22, pp. 7497-7501, Nov 2017. doi: 10.1109/JSEN.2017.2756876

[6] Yang, Won Tak, So Yeon Park, Dongjun Suh, and Seonju Chang. "LAMF: Lighting Adjustment for Mood by Food: RFID Based Context Aware Food Ordering and Lighting Control System at Restaurants." In *2013 International Conference on Information Science and Applications (ICISA)*, pp. 1-3. IEEE, 2013.

[7] Vaseloff, Dennis John, and Loren Jay Veltrop. "Smart tray system and method for restaurant inventory management." U.S. Patent 7,132,926, issued November 7, 2006.

[8] Vaseloff, Dennis John, and Loren Jay Veltrop. "Smart tray system and method for restaurant inventory management." U.S. Patent 7,132,926, issued November 7, 2006.

[9] Kimball, James F., and Stephen B. Leonard. "Inventory management system using RFID." U.S. Patent 7,680,691, issued March 16, 2010.

[10] Schackmuth, Glenn, and Gerald Sus. "RFID food production, inventory and delivery management system for a restaurant." U.S. Patent Application 11/413,234, filed November 1, 2007.