Automated Restaurant Table Management

Sree Hari Nathan K S
Department of Electronics and Communication Engineering
Amrita School of Engineering, Amrita Vishwa Vidyapeetham,
Coimbatore, India
cb.en.u4cce21070@cb.students.amrita.edu

S Nithin
Department of Electronics and Communication Engineering
Amrita School of Engineering, Amrita Vishwa Vidyapeetham,
Coimbatore, India
cb.en.u4cce21059@cb.students.amrita.edu

Vishveshwaran K
Department of Electronics and Communication Engineering
Amrita School of Engineering, Amrita Vishwa Vidyapeetham,
Coimbatore, India
cb.en.u4cce21078@cb.students.amrita.edu

Kavinvelavan M
Department of Electronics and Communication Engineering
Amrita School of Engineering, Amrita Vishwa Vidyapeetham,
Coimbatore, India
cb.en.u4cce21031@cb.students.amrita.edu

***Abstract*—ARTM (Automated Restaurant Table Management) is a model where both hardware and software components are used to manage and optimize the restaurant. ARTM is composed of software tools such as Google Cloud Platform, Spreadsheet database, Thonny IDE (Integrated Development Environment), Python, VS Code environment and hardware component used is RaspberryPi Camera module. The primary objective of this proposed project is to automate the entire customer interaction in the restaurant chain system**

Keywords—QR code, RaspberryPi Cam, queue, HTML

I. Introduction

We live in an era of rapid change whereby the dynamic, highly competitive business environment, along with everchanging customer preferences, and the constant emergence of new technologies force organizations to continuously reorganize and reinvent themselves. Research suggests that service organizations are increasingly turning to technological innovations. One such innovation is the Automated Restaurant Table Management (ARTM).

The traditional restaurant model is time consuming and tedious process where a person from the restaurant takes customer name and waits the customer till the tables are unoccupied. The problem arises when the customer queue is large and customer take to occupy seats without the notice of representative. Hence ARTM is used to fully automate the process.

II. Ease of Use

The proposed model aims to improve the ease of use for customers in a restaurant setting. By providing both hardware and software techniques the process makes it streamlined and eliminates need for physical interaction between the customer and representative and traditional way of seating. The use of ARTM makes convenience and ease use of customers. Overall, the proposed model works by prioritize of the customers present in database queue. It works by FIFO (First In First Out) process.

III. Literature Review

In the traditional restaurant model it is a manual process where a representative from the restaurant takes customer name and wait till the tables available before assigning them. This is an inefficient, time consuming and tedious process. There are several disadvantage when the queue is large. Hence ARTM is proposed to fully automate the table management system [1].

The existing literature review in traditional restaurant management highlights the limitation of manual process. Studies have shown that the long waiting time can make impact on customers. Also there is a high probability that ARTM improve the efficiency and overall productivity [2].

The ARTM (Automated Restaurant Table Management) provides efficient solution to the challenges in the traditional management. The use of technologies such as web based management , QR (Quick Response) and integration with Firebase technology manage queue and table availability [3].

The key implementation considerations for the ARTM are

- Hardware Setup
- Software Integration

A. *Hardware Setup*

The ARTM uses RaspberryPi Cam module to detect table availability using QR (Quick Response) code. There are certain measures to implement Hardware Setup. Proper placement and position of the Camera module is required to detect QR code and a reliable communication is made between the database and the RaspberryPi [4].

B. *Software Integration*

The ARTM uses software technologies such as HTML (HyperText Markup Language), CSS (Cascading Style Sheets) and JavaScript to create form to store customer data entry and accessed through database. Google Spreadsheet and Cloud Platform is used to store and accessing form data. Python script is used to manage the customer queue and update the database based on output of Hardware Setup. Firebase is used for displaying the final output [5].

IV. Methodology

We have proposed a system which generally automates the process of user table allocation in restaurants. There is a separate system placed in the restaurant for the persons who wants to dine in. The persons can show their presence for dining in by entering the details required such as the name of the person and the seater preference and which will be appended to the queue then. The queue contains the similar details of all other persons who have inputted their details before. Once a person is assigned to a table, their entry will be popped out from the queue so that the system will work with correct consideration.

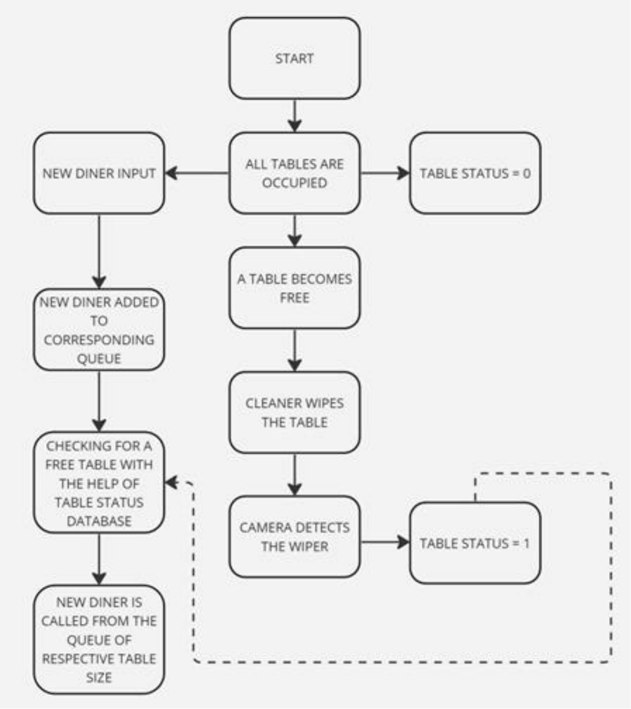

START
ALL TABLES ARE OCCUPIED
NEW DINER INPUT
TABLE STATUS = 0
A TABLE BECOMES FREE
NEW DINER ADDED TO CORRESPONDING QUEUE
CLEANER WIPES THE TABLE
CHECKING FOR A FREE TABLE WITH THE HELP OF TABLE STATUS DATABASE
CAMERA DETECTS THE WIPER
TABLE STATUS = 1
NEW DINER IS CALLED FROM THE QUEUE OF RESPECTIVE TABLE SIZE

Fig. 1. Flowchart of the System

Apart from this system, there is yet another system defined which basically works with the help of a camera and a queue consisting of the table information and to be precise, the table number and its occupancy status which is either 1 or 0 where the former denotes the table is occupied and latter denotes the table is unoccupied. The entries in the queue can be updated/changed by real-time data communication between the camera and the queue. Whenever the QR code is scanned via a camera, the initial step is to find the table number from where the QR code has been scanned and the specific entry with the same table number can be updated/modified to the status to which it is supposed to be changed [6].

The two queues, one for inputting customer details and another one for storing table status showing occupied or unoccupied are compared iteratively in a loop and once a table is changed to status 0 denoting unoccupied the next person in the queue will get assigned to that table. But in addition to this, there is another condition to be checked whether the table that is free currently and the seating preference of the next person in the queue are same and if that's the case the person will be automatically assigned to that particular table. Just in case if the condition went false, the system is made to check for next person in the queue with seater preference matching with the seating occupancy of that table got freed. Once the condition is found to be true, the person will be allocated to that particular table and the entry with the person's details will be removed from the queue and that particular table's status will be updated to 1 signifying that the table is occupied now. This is how the implementation goes.

Fig. 1 shows the complete implementation of the system. Initially, if the table is occupied, the conditions become true and table status gets assigned to 0. Alongside, input from diners will be appended into the queue in a continuous manner. Once the table is freed, the cleaner wipes the table and the QR code pasted in the wiper gets detected by the camera attached to the table. A 20 seconds delay is generally present in order for a smooth table cleaning process. Since the next person is called in for dining, the table status is modified into 1 signifying that the table is occupied.

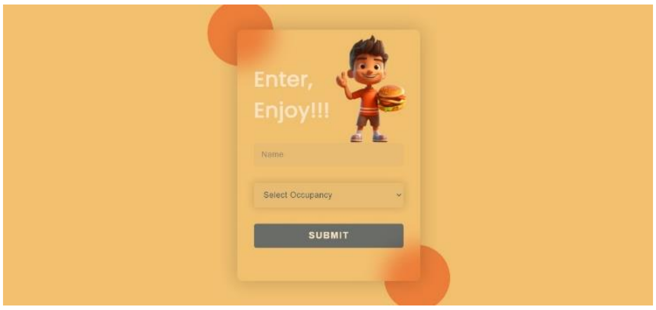


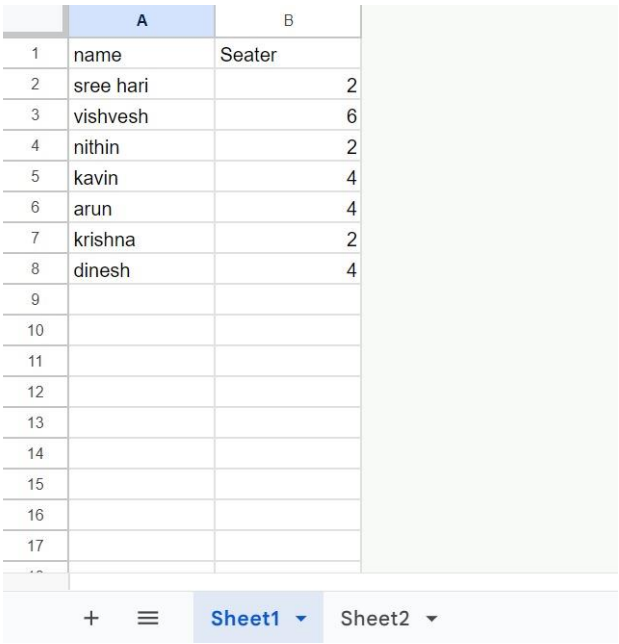
Fig. 2. User Entry Page

V. Results

Fig. 2 showcases the user interface (UI) of the customer details entry page where the person who visits the hotel for dining can input their name and the seater occupancy they need and clicking the submit button will append the entry to the queue and will be put under consideration for table allocation.

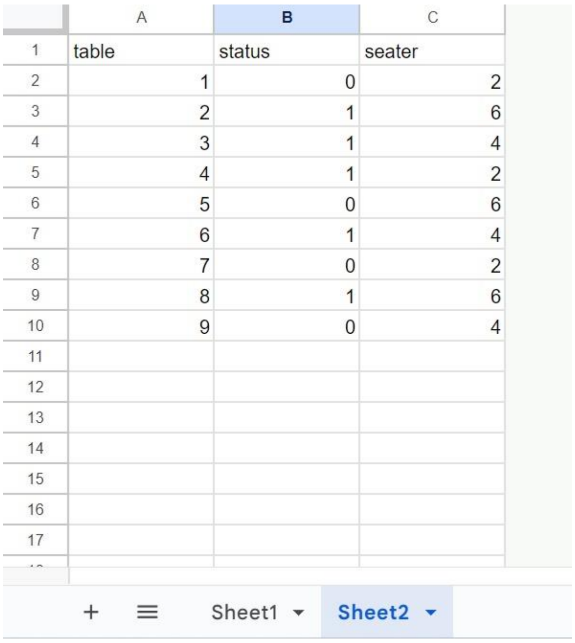
Fig. 3 tells how the queue containing the received entries looks like. The person standing at the top of the queue will be allotted next only if the condition of occupancy capacity of a table and seater preference of that respective person matches.

Fig. 4 includes the current status of tables available in the restaurant. The first column contains the table number, second column is the occupancy status of the table being 1 or 0 where 0 signifies that the table is occupied and 1 denotes it is unoccupied. The last column includes the occupancy of the respective table.



	A	B
1	name	Seater
2	sree hari	2
3	vishvesh	6
4	nithin	2
5	kavin	4
6	arun	4
7	krishna	2
8	dinesh	4
9		
10		
11		
12		
13		
14		
15		
16		
17		

Fig. 3. Customer Queue



	A	B	C
1	table	status	seater
2	1	0	2
3	2	1	6
4	3	1	4
5	4	1	2
6	5	0	6
7	6	1	4
8	7	0	2
9	8	1	6
10	9	0	4
11			
12			
13			
14			
15			
16			
17			

Fig. 4. Table Status Queue

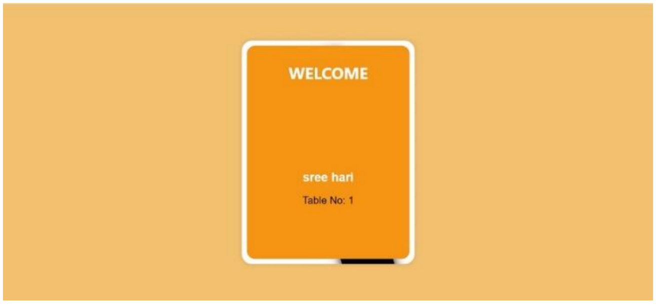
Fig. 5 demonstrates how the person will be called in once a table gets freed. Firebase has been used for this implementation. Once the table gets freed, the table number and the name of the person to whom the table has assigned will be sent to firebase integrated with python. Since it belongs to display side, HTML and CSS has been used to display the information. From firebase, the data has been fetched and send to HTML [7].

VI. Limitations

One of the main limitations of our project is that we need a camera for each table to detect the customer's presence in that particular table. This increases the cost of the objective and if the restaurant increases its capacity, then new cameras have to be bought separately. This means scalability is also a major limitation in our project. There are many other concerns like privacy of the customer, maintenance of the cameras, etc.

VII. Conclusions

On the whole our project, ARTM, helps in increasing efficiency and accuracy of table management in busy restaurants. It also helps in freeing up staff for other tasks and maintains the fairness of table allocation. The chances of missing that a table got freed is very low in ARTM compared to a human who is monitoring for a free table. Although our project has its own limitations like any other project, it can be made efficient on further optimizations. To conclude, this project paves a way for the future of restaurant table management making the process easier to both customers and restaurant owners.



WELCOME

sree hari

Table No: 1

Fig. 5. Output Display

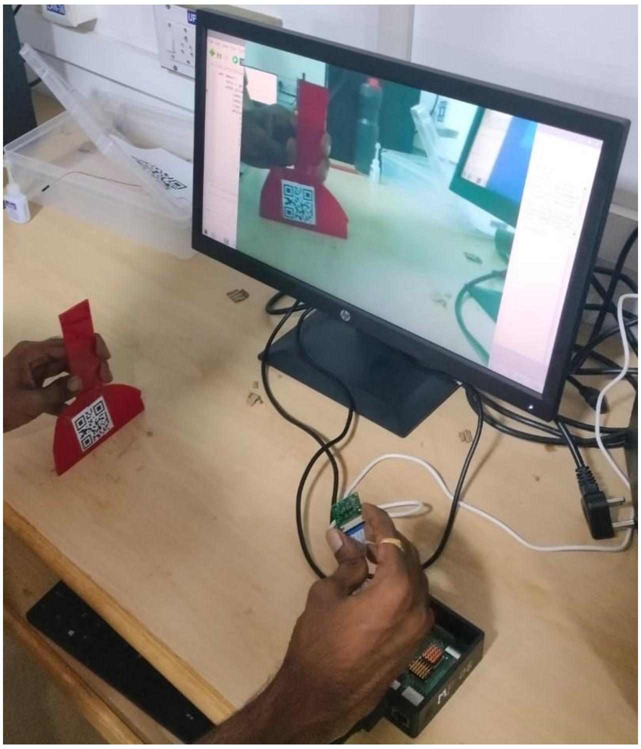


Fig. 6. Live Implementation

VIII. References

[1] "(5) (PDF) Automated System for Restaurant Services." Accessed: Jun. 05, 2024. [Online]. Available: https://www.researchgate.net/publication/356894797_Automated_System_for_Restaurant_Services

[2] C. H. Tekawade, R. G. Pathade, S. A. Jadhav, and R. D. Yadav, "Automated Restaurant Management System," *IJRASET*, vol. 11, no. 5, pp. 6703–6706, May 2023, doi: 10.22214/ijraset.2023.53195.

[3] H. Saeed, A. Shouman, M. Elfar, M. Shabka, S. Majumdar, and C. Horng-Lung, "Near-field communication sensors and cloud-based smart restaurant management system," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, USA: IEEE, Dec. 2016, pp. 686–691. doi: 10.1109/WF-IoT.2016.7845440.

[4] H. Ghael, "A Review Paper on Raspberry Pi and its Applications," Jan. 2020, doi: 10.35629/5252-0212225227.

[5] M. A. Mokar, S. O. Fageeri, and S. E. Fattoh, "Using Firebase Cloud Messaging to Control Mobile Applications," in *2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, Khartoum, Sudan: IEEE, Sep. 2019, pp. 1–5. doi: 10.1109/ICCCEEE46830.2019.9071008.

[6] C. Patel, A. Patel, and D. Patel, "Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, vol. 55, pp. 50–56, Oct. 2012, doi: 10.5120/8794-2784.

[7] D. S. Blyth, "Web page design for HTML (and friends)," in *IPCC 96: Communication on the Fast Track. IPCC 96 Proceedings*, Saratoga Springs, NY, USA: IEEE, 1996, pp. 89–103. doi: 10.1109/IPCC.1996.552585.