2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4) | 978-1-6654-0096-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/WorldS451998.2021.9514018

Service Robots in Hospitals To Reduce Spreading of COVID-19

Hussain Dadi
Dept. of Computer Science andEngineering
College of Engineering, Qatar University
Doha, Qatar
hd1404045@student.qu.edu.qa

Mubarak Al-Haidous
Dept. of Computer Science and Engineering
College of Engineering, QatarUniversity
Doha, Qatar
ma1515017@student.qu.edu.qa

Nayef Radwi
Dept. of Computer Science andEngineering
College of Engineering, Qatar University
Doha, Qatar
nr1707337@student.qu.edu.qa

Loay Ismail
Dept. of Computer Science and Engineering
College of Engineering, Qatar*University*
Doha, Qatar
loay.ismail@qu.edu.qa

***Abstract*— COVID-19 is an infectious disease that is introduced by a new coronavirus. The COVID-19 virus symptoms can include coughing, fatigue, fever for those with mild cases. For more severe cases, it can lead to shortness of breath, chest pain, and eventually death. Due to its infectious nature, people started taking health precautions that were advised by the World Health Organization such as social distancing, washing hands, and sanitizing surfaces. Due to this pandemic, the aim of this project is to automate mundane tasks with the help of a service robot to reduce physical contact as well as reduce the workload on healthcare workers. Tasks such as delivering food, medication, water, and other necessities to patients with infectious diseases. In addition, this project specifically reduces physical contact between patients and the attending nurse. A service robot is a robot that is built to perform a repetitive or dangerous task in order to assist humans. A service robot executes repetitive tasks more efficiently compared to a person who is prone to lose focus and make mistakes during work hours. The service robot performs tasks that are requested by the patients and accepted by the nurse through the mobile application. The application shows both the nurse and the patients the position of the robot and the task it is performing. The project's design was based on a small robot car that receives a path from the server upon a request using a mobile app. On top of that, the project includes a functional mobile application, a RESTful API, a no-SQL database, and a python microservice that is responsible for path planning. The path planning algorithm used is the A* algorithm for finding the shortest path to the destination. The service robot avoids obstacles by first stopping after detecting the obstacle, label the coordinates of that location as obstacle, rotates 180 degrees, moves to the previous location, and then requests the path again. The constant communication between the robot and the server allows the server to track the robot's movement.**

Keywords—service robot, COVID 19, microservice, Arduino, Raspberry Pi, RESTful API, Mobile Application, A* Algorithm, Autonomous Robot

I. INTRODUCTION

COVID-19 has impacted the world and the lives of millions in every aspect. It changed how humans interact with one another, and it changed how humans communicate with one another. In such outbreak, the WHO (World Health Organization) works toward finding a vaccine and provide guidelines and rules to follow to reduce the risk of catching a Virus. However, it is also the duty of humanity in general to help counter the outbreak and figure out ways to control the pandemic, especially in hospitals. According to the Centers for Disease Control and Prevention (CDC), roughly 1.7 million hospital-associated infections contribute to around 99,000 deaths each year, this is only in the United States of America [1]. COVID-19 can be transmitted from person to person in household, or in any healthcare center. The majority (64%) of cases were between the age of 25 to 64 years according to WHO [2]. However, the mean age of death is 80. In such circumstances, attending nurses to those elderlies need to be of higher priority than those of younger age because of the lethality rates among the elderly. Then the question becomes: since the number of healthcare workers is in utmost need during such pandemic, is there a way to have the workers focus mostly on the patients who are on the brink of death, while still giving necessary attention to other patients without physical contact? The answer can be provided with the help of service robots. A service robot is an autonomous robot capable of assisting humans by performing tasks that are repetitive and dangerous to humans. The use of a service robot in health industries can reduce the risk of infecting healthcare workers and help in tasks such as delivering foods, water, medication, etc.

According to the Ministry of Public Health, there have been more than 200000 confirmed cases of coronavirus in Qatar [3]. Healthcare workers were at utmost demand. One of the main reasons why automating the delivery of necessities such as food, medication, and water is the reduction of the spread of viral infection. Robots are immune to infections, and do not require food and exercise to work well; feeding it electrical power is all that it needs to operate. According to Bill Huang, who is the chief executive of Cloud Minds, a Chinese–US company that provides a cloud-based robot operating system, on 28 February China designed the first robot-run ward that is used to prevent staff at Wuhan Wuchang Hospital from contacting with COVID-19. By 6th of March 2020, they have seen a sharp drop in the cases of the virus [4]. The drop in cases could be explained by the preparedness of a country to counter the outbreak but also robots played a huge role in the reduction of the virus spread in hospitals. However, robots will prevent infection form a person to others not only for COVID-19 outbreak, but also for future unpredictable viral infections that could sweep around the globe. In Qatar, since the beginning of the COVID-19 outbreak, 13000 nurses have provided services to tens of thousands of patients

[5]. These nurses had to deal with the stresses and challenges to provide all the necessary attention to patients with the COVID-19 and patients with other conditions. With this huge deployment of nurses, it is only reasonable to be able to design a robot capable of performing some of the tasks for the nurses.

Designing a robot that can work in hospitals from distance and be able to deliver items to patients while also disinfecting surfaces and measuring temperatures is highly effective for both the prevention of COVID-19 outbreak and saving time for healthcare workers.

The reason for attempting to complete the project is to benefit the people of Qatar, and the world to reduce the transmission of COVID-19 virus. Also, the need to help others is what got us into this project. In section 2, the paper illustrates the system design overview with a scenario about the flow of the system. Followed by section 3, the Hardware design with its hardware architecture which describes the main hardware components with their connections. Section 4 consists of the description of the software design and the role of each part. Results are in section 5, it shows the results of the testing performed on the key constraints for both the hardware and the software. Finally, the conclusion summarizes the work presented in the paper and the future work.

II. System Design

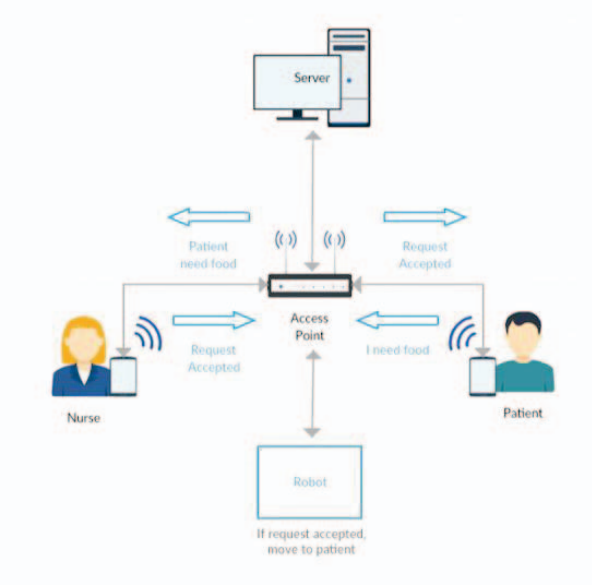

Fig. 1. System's block diagram.

Assume the following scenario: A patient wants to order food and water from a nurse in a specific floor. The patient can request the service of food delivery to the server via mobile application. The user-interface of the mobile application consists of a real-time map of the floor of the hospital and the robot's position. After sending the request to the server, the nurse -using the same mobile application- can have the option of viewing the list of requests from patients and choose to either accept or discard the request. The reply is sent back to the server with the nurse's choice. If the request is accepted, the server then starts the path planning algorithm, which finds the shortest path from the starting position of the robot to the destination, and issues 4 basic commands: FORWARD – BACKWARD – LEFT – RIGHT according to the path acquired from the path planning algorithm. In response, the robot receives the commands and operates. If the robot encounters dynamic obstacles like people along its path, the robot simply stops and wait for 3 seconds. If the obstacle did not move, then it recalculates the best path again. However, static objects will not be counted in the obstacle avoidance algorithm because they already exist as part of the map itself. After reaching the destination, which is the patient room, the robot scans the QR code on the patient's bed to match it with its destination for verification. Upon completion, the robot marks the request as a success and returns to its initial position.

III. Hardware Design

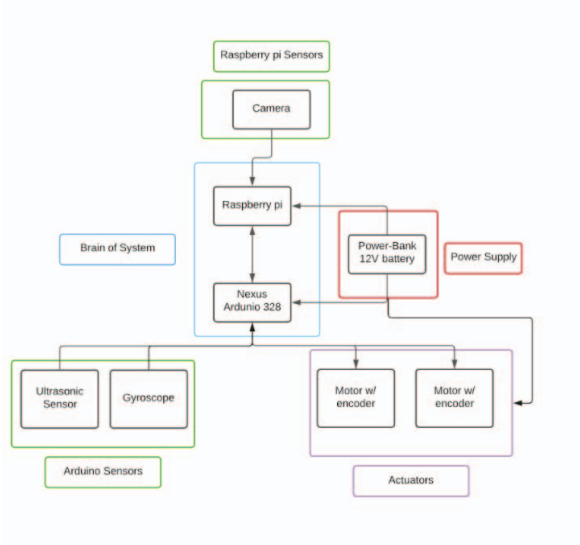

Fig. 2. Hardware Architecture.

This block diagram shows the main hardware components as well as the connectivity between the components. The components split into 4 main categories: sensors, actuators, brain of system, and power supply.

A. Brain of system

The hardware of the brain of system has both the Arduino and the Raspberry pi use bidirectional communication. The Raspberry pi sends the commands to the Arduino, and the Arduino sends a confirmation string in case it moved a given distance, and an alert string in case the robot encounters a dynamic obstacle. The reasons behind using two microcontrollers are mentioned in Hardware design section.

B. Sensors

Each sensor serves a specific function, and the sensors are used to feed the brain of the system with specific values that it uses for decision making. Sensors are split into 2 categories: Arduino sensors, and Raspberry sensors.

Arduino sensors: The first sensor is the gyroscope sensor which works along with the wheel encoders to sense the robot's orientation and distance moved. Also, it consists of an ultrasonic sensor which is used to detect obstacles in the path taken by the robot, and a current-voltage sensor capable of measuring the

current and using an algorithm that estimates how much of the battery is left. This is useful in case if the battery is near depletion and decide to not make a delivery until it is recharged.

Raspberry pi sensors: The camera is used to detect and decode a QR code. The QR code is used to help the robot identify its start and destination. It is also responsible for directing the commands to the robot after being processed.

C. Actuators

When it comes to the actuators, the 2 Faulhaber 2342motors are driven by a built in motor driver inside the Arduino are used to move and rotate the robot by the Raspberry Pi's command. With the motor's encoder, they transmit the encoded signals to the Arduino and calculate how much did the robot travel. This is very important for the mapping because in order to know where the robot is, the Server needs to update the location based on how far the robot moved.

D. Power Supply and Battery

The whole system runs on a 12V battery capable of delivering 2.5A for the motors, as well as provide 5V to the raspberry pi and the Arduino. Assuming the robot runs at full speed, the power required for the system will be roughly 48 watts.

E. The Robot

The robot was designed to withhold heavy weight and move in different environments and terrains. 31cm x 30 cm x 11cm, this robot dimension enables the use of this robot in many places. It is a simple, tracked mobile robot to program with Arduino IDE as they build a board specifically to be run by the two motors operating inside the robot. The motors are 12V geared DC motor Faulhaber 2342 with 17W motor power capable of holding a load up to 20 kg. The whole-body material is made from tough Aluminum Alloy that can withstand heat and has light weight. A power button and a charger are placed at the front of the robot to ease the user's experience.

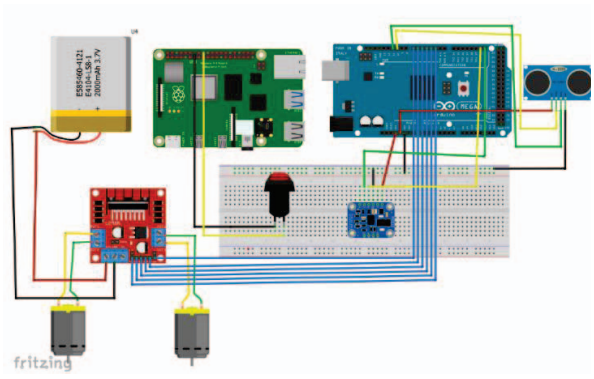

Fig. 3. Electric wiring diagram.

Fig. 5 shows how the software, data flow, and the communication of the proposed solution are interconnected with each other as well as it gives a clear indication of how the whole system works from a software point of view.

The diagram consists of 4 main parts, each part has a key role in the proposed solution and their collaborative work is necessary.

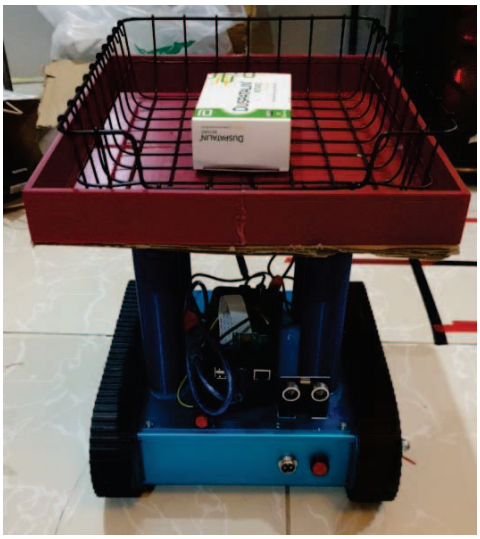

Fig. 4 Robot system.

IV. Software Design

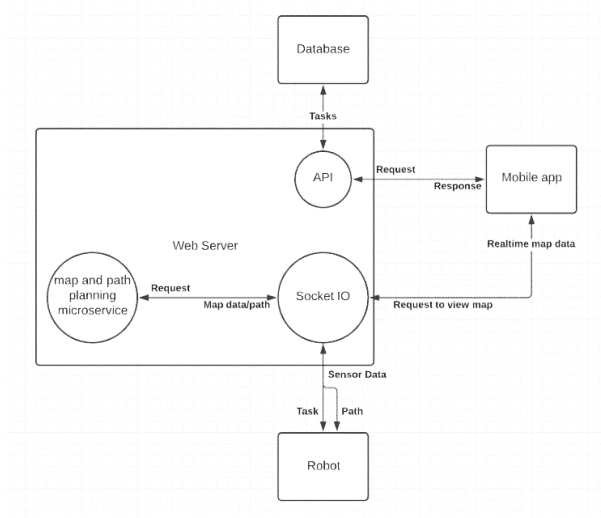

Fig. 5 Software Architecture.

A. Mobile App

The first part is the mobile app, which allows the user to send tasks to the server, view tasks, accept or decline tasks, and view a real-time map that indicates the robot's position. All these options are divided in two types of clients: nurse and patient. The nurse is able to view tasks and accept or decline tasks, while the patient is able to send tasks in the form of a request. Both users are able to view the map.

B. Server

The second part of the system is the server, the server consists of 3 sub-parts that are responsible for processing and redirecting data between other vital parts of the system. The API handles the user's requests as well as the communication with the database. The socket IO service is responsible for transmitting or receiving real-time data that requires urgent processing and are not delay tolerant such as the robot's movement on the real-time map. The path planning algorithm guides the robot by producing a sequence of steps that the robot follows. The path planning algorithm used is A* algorithm to find the shortest path from the robot's position to the destination [6]. The algorithm used as a separate microservice that is contacted by the webserver. This microservice also creates the image of the map that has the robot's position. The third key component of the block diagram is the database, which is used to store user's data and the requests made to the server. The data can be fetched by the API and served to the mobile application.

The server is built using an API and a microservice. The API is built using NodeJS[7], MongoDB[8], Socket.IO[9], and Express framework[10] which uses JavaScript. The microservice is programmed using FastAPI, cv2, NumPy, and Matplotlib which are python frameworks and libraries.

The API is the core of the integration between the robot, the mobile app, and the microservice. The express API routes are used for the mobile app to make requests that are related to creating, editing, fetching, and deleting patient's orders as well as order acceptance by the nurse. These orders are stored using MongoDB, which is a No-SQL database. The socket IO is responsible for streaming the map and receiving data from the robot. The Socket IO part of the API is split into events and rooms that allow data separation; this separation is used to ensure that clients receive the correct data. An example of this is a map base-64 encoding reaching the correct patient that is on the floor of that map.

C. Database

The database plays an important role in making the application consistent by ensuring that the data is safely stored and can be easily retrieved. Information such as patient orders, nurse details, and hospital floor maps that are retrieved from the database to find the shortest path from the robot's charging station to the patient's room.

V. Results

A. Accuracy

Testing the accuracy is necessary to ensure that the robot is moving the correct distance of 30 cm. The test is done by making the robot move for the expected distance of 30 cm, then measuring the actual distance moved by the robot using a measurement tape.

This process was repeated 10 times. With the collected data, a table, which is created using Google spreadsheets, was created to show multiple parameters, including the error ((measured-expected)/expected * 100), average of distance, average of error, minimum distance, and maximum distance.

In conclusion, it is verifiable that the average error is around 3.63 percent and that this error can be mitigated by using quality encoders and decrease the sampling time below 300ms. The error however cannot be completely solved as there will always be slight differences between each individual motor even after using a PID controller. The test shows that the robot partially met the design constraint of an accuracy of 30 cm.

TABLE I. DISTANCE DATA TABLE

Trial	Distance (cm)	Error (%)
1	30.0	0.00
2	32.0	6.67
3	29.8	0.667
4	32.8	9.33
5	32.0	6.67
6	30.0	0.000
7	30.0	0.000
8	32.8	9.33
9	31.0	3.33
10	30.5	1.67
Average	31.09	
Average Error	3.63 %	
Min Distance	29.8	
Max Distance	32.8	

B. Obstacle Detection

A map was created to do the test of obstacle detection. The map is a 1.8m x 1.8m divided into 6 x 6 squares, each having a dimension of 30cm x 30 cm. The figure below illustrates the map that was designed for testing the obstacle detection, and what will be used for testing the integration.

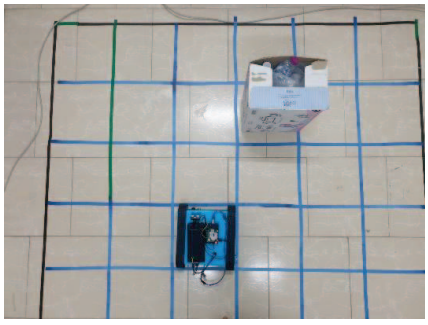

Fig. 6 floor map with obstacle.

To sum up table II, the distance average was around 12.36 cm, and the average error is 18.2 percent. The reason for the difference in reading is because of the different orientation of the obstacle. Changing the obstacle's orientation means that the

distance from the ultrasonic changes because the edge of the box is closer to the sensor. Also, another reason is the sampling time of detection, it is set to 300ms. By reducing the sampling time, it is possible to achieve lower error degrees. It is worth mentioning that the robot did not collide with the obstacle and has met the practical design constraint for safety.

Table 3 shows the results of the performance of the server. The server was tested on 2 criteria, how much memory is being used and the response time.

TABLE II. OBSTACLE DETECTION DISTANCE DATA TABLE

Trial	Distance (cm)	Error (%)
1	15.5	3.33
2	12.0	20.0
3	8.00	46.7
4	12.0	20.0
5	13.0	13.3
6	13.1	12.7
7	13.0	13.3
8	15.0	0.00
9	14.0	6.67
10	8.00	46.7
Average	12.4	
Average Error	18.3 %	
Min Distance	8.00	
Max Distance	15.5	

TABLE III. SERVER PERFORMANCE

Performance criteria	Min	Max	Average
Memory usage	46 MB	57.3MB	49.3MB
Response time	18ms	831ms	227ms

C. The User Interface

The mobile application is implemented using Flutter[11], which is a software development kit made by Google and uses Dart as its primary programming language. Flutter is used to build cross-platform applications that can work on web, desktop, IOS, and Android. For this project, the application supports only IOS and Android out of the available platforms. The mobile application includes the implementation of a login screen for both nurses and patients, a main screen that handles orders for both nurses and patients, a real-time map screen, and a settings screen. Fig. 7 shows the implementation of the mobile application, the screens blow are the patient's main screen, the real-time map screen that is accessible by patients and nurses, the create new order dialog that is accessible by the patient, and finally the nurse's main screen that allows the nurse to accept or reject orders.

VI. CONCLUSION

To sum up the results of the work done, the project fulfills the objectives set as expected. The robot is able to deliver or pickup orders that are received from the patient and by doing so the robot reduces nurse and patient contact. The users can view a map of the robot's position and the robot receives the shortest path from the server to execute orders. The main functionalities of the robot are working correctly and have been implemented in a scalable way allowing a lot of space for improvements. This work acts as a strong steppingstone to a larger application in the future, where multiple robots can operate collaboratively at the same floor.

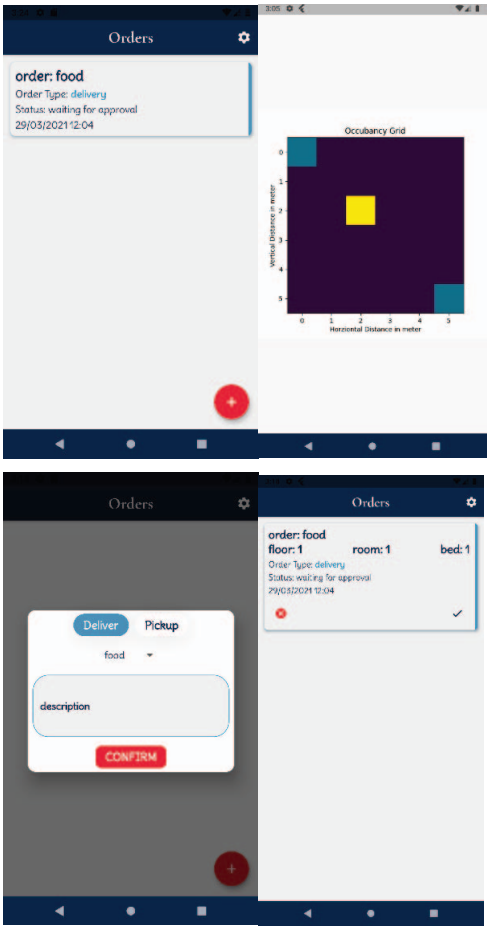

Fig. 7 Mobile Application User Interface.

REFERENCES

[1] R. M. Klevens *et al.*, "Estimating health care-associated infections and deaths in U.S. Hospitals, 2002," *Public Health Rep.*, vol. 122, no. 2, pp. 160–166, 2007, doi: 10.1177/003335490712200205.

[2] World Health Organization, "Coronavirus disease (COVID-19) Situation Report – 198," *A A Pract.*, vol. 14, no. 6, p. e01218, 2020.

[3] COVID19 Home. [Online]. Available: https://covid19.moph.gov.qa/EN/Pages/default.aspx. [Accessed: 17-May-2021].

[4] S. O'Meara, "Meet the engineer behind China's first robot-run coronavirus ward," *Nature News*, 24-Jun-2020. [Online]. Available:

https://www.nature.com/articles/d41586-020-01794-8. [Accessed: 12-Nov-2020].

[5] "13,000 Nurses at the Forefront of Care for COVID-19 Patients." https://www.hamad.qa/EN/news/2020/July/Pages/13000-Nurses-at-the-Forefront-of-Care-for-COVID-19-Patients.aspx (accessed Nov. 12, 2020).

[6] I. Zidane, K. Ibrahim, "Wavefront and A-Star Algorithms for Mobile Robot Path Planning," in International Conference on Advanced Intelligent Systems and Informatics, Cairo, EG, 2018, pp.69-80.

[7] "Node.js v16.1.0 documentation," Index | Node.js v16.1.0 Documentation. [Online]. Available: https://nodejs.org/api/. [Accessed: 17-May-2021].

[8] "The MongoDB 4.4 Manual" The MongoDB 4.4 Manual - MongoDB Manual. [Online]. Available: https://docs.mongodb.com/manual/. [Accessed: 17-May-2021].

[9] D. Arrachequesne, "Server API," Socket.IO, 15-May-2021. [Online]. Available: https://socket.io/docs/v3/server-api/index.html. [Accessed: 17-May-2021].

[10] "5.x API," Express 5.x - API Reference. [Online]. Available: https://expressjs.com/en/5x/api.html. [Accessed: 17-May-2021].

[11] "Flutter documentation," Flutter. [Online]. Available: https://flutter.dev/docs. [Accessed: 17-May-2021].