# Self-driven BumbleBot for Medicine Delivery in Hospital Environment

by

Syed Reyadh

Student ID: 19035702

Supervised by: Dan Withey

Date: 24/04/2023

Department of Engineering Design and Mathematics

Module Name: Individual Project BEng

Module Code: UFMFX-8-30-3

Award Programme: Robotics

# Table of Contents


List of Figures: 3

List of Tables: 4

Disclaimer 4

Acknowledgement 4

Abstract 5

1.Introduction: 6

2. Aims and Objectives: 7

    2.1 Aims: 7

    2.2 Objectives: 7

    2.3 Research Questions: 7

    2.4 Project Scope: 7

3. Project Management 9

    3.1 Contact Register and Logbook: 10

    3.2 Risk Awareness and Mitigation: 10

    3.3 Work Breakdown Structure: 12

    3.4 State-machine: 13

4. Context to the Work 14

    4.1 Background Literature Review: 14

    4.2 Some Current Applications of Autonomous Robot in Hospital Environments: 15

    4.3 Interpretation of Previous Work in the Topic: 17

    4.4 Critical Discussion of Relevant Published Work: 19

5. Research Methodology 21

    5.1 Requirement Analysis: 21

    5.2 Virtual Environment Design: 22

    5.3 Bumblebot Robot with its URDF 23

    5.4 Robot Navigation and Localization: 36

    5.5 Bumblebot's Task Planner 38

    5.6 Project Limitations: 42

6. Results: 43

    6.1 Completed Virtual Environment: 43

    6.2 Accomplished Robot: 44

    6.3 Mapping Completion: 45

    6.4 Bumblebot's complete operation: 46

7. Analysis and Scientific Argument: ........ 51
7.1 Development and Coherence of Arguments from the literature: ........ 51
7.2 Comparative analysis between project findings and literature review: ........ 51
7.3 Development and quality of the scientific argument: ........ 51
7.4 Evidence of the ability to evaluate information and synthesise conclusions: ........ 52
7.5 Innovation style: problem solving, or step-by-step learning through established techniques: ........ 52
8. Evaluation and Accomplishment: ........ 53
8.1 Critical appraisal and evaluation of the project and process: ........ 53
8.2 Reflection of self-development whilst conducting the project: ........ 53
8.3 Achievements and shortcomings of the project: ........ 54
8.4 Further development and recommendations: ........ 55
8.5 Relating the project to UK-SPEC competencies: ........ 55

## List of Figures:

Figure 1: Updated Project Gantt Chart........08
Figure 2: Work Breakdown Structure........11
Figure 3: Project State-Machine........12
Figure 4: The Tug Autonomous Medical Robot........14
Figure 5: Panasonic Autonomous Delivery Robots........15
Figure 6: WDR01C Model........16
Figure 7: Turtlebot in research virtual environment........17
Figure 8: Turtlebot started navigation........18
Figure 9: Concept Hospital Environment Sketch........21
Figure 10: Created 3D Hospital Environment in SweetHome3D........22
Figure 11: Concept Robot........22
Figure 12: Bumblebot's all links........23
Figure 13: Bumblebot's all joints........24
Figure 14: left_wheel link and joint in URDF........25
Figure 15: right_wheel link and joint in URDF........26
Figure 16: caster_front link and joint in URDF........27
Figure 17: caster_back link and joint in URDF........28
Figure 18: lidar link and joint in URDF........29
Figure 19: basket link and joint in URDF........30
Figure 20: arm_base link and joint in URDF........31
Figure 21: link1 link and joint in URDF........32
Figure 22: link2 link and joint in URDF........33
Figure 23: link3 link and joint in URDF........34
Figure 24: Hand drawn path plan........35
Figure 25: Robot has started exploring by using Gmapping........36
Figure 26: Robot is about to start navigation........36
Figure 27: Flowchart for robot task programming........37
Figure 28: Necessary Packages........38
Figure 29: Goal Function........39

Figure 30: Carry Function……………………………………………………………………39
Figure 31: Main Function………………………………………………………………………41
Figure 32: Completed Virtual World…………………………………………………………..43
Figure 33: Completed Bumblebot………………………………………………………………44
Figure 34: Completed Map………………………………………………………………………45
Figure 35: Step-1…………………………………………………………………………………46
Figure 36: Step-2…………………………………………………………………………………46
Figure 37: Step-3…………………………………………………………………………………47
Figure 38: Step-4…………………………………………………………………………………47
Figure 39: Step-5…………………………………………………………………………………48
Figure 40: Step-6………………………………………………………………………………...48
Figure 41: Step-7………………………………………………………………………………...49
Figure 42: Step-8………………………………………………………………………………...49
Figure 43: Step-9…………………………………………………………………………………50

## List of Tables:

Table 1: Project Scope……………………………………………………………………………………………………….08

Table 2: Risk Assessment…………………………………………………………………………………………………….10

Table 3: Objective Completion……………………………………………………………………………………………..54

## Disclaimer

In submitting this assignment, I make the following declaration:

- I declare that I am the sole author of this work.
- I have not copied work from any source (including my own previously submitted work for which credit has been/is due to be awarded at UWE or elsewhere).
- I have not shared any versions of my work being submitted with other students.
- I have not viewed any versions of work being submitted by other students.
- I have fully acknowledged/referenced all sources of information used.
- I am aware that failure to comply with the above may constitute an assessment offence.

## Acknowledgement

"I wish to express my thanks to the project supervisor, Dr. Dan Withey, for his support and mentoring throughout the process of conducting this study."

# Abstract

The goal of this project was to mimic a hospital environment using ROS, Gazebo, and Rviz, and then design and control a virtual robot to carry out various duties within the environment. The project started by building a virtual hospital environment in SweetHome3D and importing it into Gazebo. A navigational map was then made using SLAM gmapping, and the robot is manufactured and customised. The robot is programmed to carry out duties including travelling to particular spots and picking up and delivering goods to predetermined destinations. Software like Rviz and Gazebo are used to keep track of its localization and mapping.

The accomplishment of the tasks and the robot's aptitude for accurate navigation were used to gauge the project's success. The project's results showed how ROS, Gazebo, and Rviz can be used to make a realistic and interactive medical environment, as well as how a simulated robot could be built to carry out various duties in the environment. The project emphasised the value of simulation technology in robotics research and development as well as its potential applications in a number of different fields, including healthcare.

The idea can be expanded through future study by including more difficult tasks and difficulties for the robot, like obstacle avoidance and course planning. A physical robot that can carry out tasks in a real hospital environment is one example of how the project might be expanded for use in the real world.

# 1.Introduction:

In recent years, the demand for development of autonomous robots for various applications in healthcare has significantly increased. Medication administration is one of the most promising fields because it involves precision, effectiveness, and dependability to guarantee patient safety and wellbeing.

According to a recent analysis from the Health Foundation, NHS worker numbers are failing to keep up with demand, and workforce levels in important areas such as primary and community care, nursing, and mental health are continuing to deteriorate. In July 2020, the number of nurses and health visitors working in community health services fell by 1.2 percent (540 FTE workers) compared to a year earlier, continuing a long-term downward trend. (“NHS staff shortages put long-term vision for primary and community care at risk,” 2022). So, staff shortage around hospitals is a long-term problem. The outcome from the chosen project will be a ROS operated navigation robot which will be able to deliver medicine to the patients autonomously. It will be specifically made for the hospitals to get cope with the staff shortage problem.

This report includes the following:

An Aim and Objective section which explains identification of relevant investigation / research topic, realistic and challenging aims and objectives, identification of depth and breadth of the project and project scope.
Next, in the Project Management section of this report a project Gantt Chart, a logbook, a contact register, a risk assessment and mitigation of the risks, project work breakdown structure and a state machine were discussed. All the above discussion on project management site provides self-directory progress and evidence of project planning and appropriate use of supervision.
Next, in Context to the Work section background Literature Review and Research of Some Current Applications of Autonomous Robot in Hospital Environments, interpretation of previous work in the topic and critical discussion of relevant published work on this field were discussed.
After that, on Research Methodology section appropriate selection of research methods demonstrating an understanding of alternative approaches, relevant technical depth and breadth, identification, use and justification of appropriate techniques to gather and analyse data and limitations: Ethical, Environmental, Financial, Time, Policies and Human Resources were explained.
Then, on the Result section data collection and analysis based on justified methodology as well as evaluation and interpretation of discoveries were explained.
Next, on the Analysis and Scientific Argument part the development and Coherence of Arguments from the literature, comparative analysis between project findings and literature review, development and quality of the scientific argument, evidence of the ability to evaluate information and synthesise conclusions and critical appraisal of the research methods used were demonstrated.
Last but not the least on the Evaluation and Accomplishment section critical appraisal and evaluation of the project and process, reflection of self-development whilst conducting the project - reflection of problem solving skills, achievements and shortcomings of the project in relation to explicit aims and other criteria as appropriate, relating the project to UK-SPEC competencies, with discussion about wider social / industrial implications, such as ethics, environment, finance, etc. and further research, development and recommendations were explained.

# 2. Aims and Objectives:

## 2.1 Aims:

The aim of this project is to design, create, and test a dependable and effective robotic system capable of providing patients with medication and supplies in a hospital or clinical setting. The implementation of enhanced robot operating system will make the robot capable of navigating across dynamic and complicated settings while avoiding impediments. By lowering prescription errors and increasing the effectiveness of medical procedures, the project also intends to improve patient outcomes and safety.

## 2.2 Objectives:

- Creating a dependable, effective autonomous robot that can transport supplies and drugs in a clinical or hospital setting.
- Designing a user-friendly interface so that medical staff may easily configure, monitor, and control it.
- Make sure the software is reliable and able to survive frequent use in healthcare environments.
- Adding cutting-edge sensors and algorithms to improve navigation, obstacle avoidance, and responsiveness to changing situations.
- Promoting the acceptance and further development of autonomous robots in healthcare by disseminating research findings and suggestions.
- The Bumblebot robot has to be designed and built within seven months starting in October 2022, with regular progress updates and milestones set throughout the project timeline.

## 2.3 Research Questions:

Given the situation and the determined research issue, the ensuing research questions will be examined. These inquiries have been made to guide the inquiry and create a more thorough understanding of the subject. The research questions are:

- What is the problem to be solved?
- Who cares about this problem and why?
- What have others done?
- What designs and methods can be employed to develop an affordable, yet reliable navigation robot powered by Ros for medicine delivery without compromising its performance and effectiveness?
- How can the implementation of a Ros robot assist the identification of a goal and provides desired service to the patients?
- What role robots in hospital environment plays in terms of serving the patients and relevant stuffs.
- What will be this project's lifecycle?
- What is the self-driven Bumblebot's main societal impact on communities and cultures when it is used to provide medications on its own?
- How the society will be benefited from this project.

## 2.4 Project Scope:

| | | | |
|---|---|---|---|
| **Project Name** | Self-driven BumbleBot for Medicine Delivery in Hospital Environment | | |
| **Date of the project initiation:** | 01/10/22 | **Date of the project termination:** | 25/04/23 |
| **Scope Description** | <u>In Scope:</u> | | |

| | |
|---|---|
| | The final product will be a self-driven simulated delivery robot which has to do the task of delivering medicine to the patients of multiple hospital wards. All the functionalities will be done autonomously.<br><u>Out of Scope:</u><br>Create a fully developed product that is ready to sell and be used publicly. |
| **Project Deliverables** | 1. A software based simulated robot which can deliver medicine autonomously on a simulated world.<br>2. Personalize URDF file for the robot.<br>3. Supporting code. |
| **Business objectives** | 4. The project is primarily aimed towards the healthcare sector, specifically the National Health Service (NHS). It is mainly being developed for the hospital environment and it can cope with the circumstance very well. |
| **Acceptance Criteria** | 5. <u>Functional requirements</u>: The robot must be able to deliver medicines to all the beds of a ward and then return to its starting position once the work is done.<br>6. <u>User interface:</u> The robot must have a user interface that is easy to understand and use.<br>7. <u>Performance:</u> Certain performance criteria, like response speed and uptime, must be met by the robot.<br>8. <u>Information accuracy</u>: The robot must be able to provide accurate performance while delivering. |
| **Constraints** | 9. Completing the whole project in a short period of time might not be possible.<br>10. The robot sometimes might face issues while carrying objects.<br>11. The created map might not work sometimes.<br>12. The robot might not be able to identify objects in front of it while working may cause unwanted collision.<br>13. Computer sometimes might face graphics related issues. |
| **Assumptions** | 14. Robot created for hospital settings that can go through hallways and elevators while avoiding obstructions.<br>15. Robot accurately distributes medication and supplies, increasing patient safety and decreasing workload.<br>16. Extensively tested for usefulness and safety in a hospital setting.<br>17. Advanced algorithms and sensors are used in navigation and mapping to adapt to changing environments and prevent collisions.<br>18. To match design and functionality with facility and staff requirements, getting some feedbacks from healthcare professionals is a plus point. |
| **Benefits** | 19. It’s an original idea which also has a useful and clear objective.<br>20. 24/7 operation and availability.<br>21. Reduce labor costs.<br>22. Quick and accurate services.<br>23. Will be smart enough to explore itself if it gets stuck at any point. |

# 3. Project Management

The project was carried out over the course of seven months, from October 2022 to April 2023, and it was planned and managed in accordance with the initial project Gantt chart. This was created at the project's early idea stages; it offers a breakdown of the various activities needed to finish the project successfully and provides estimated completion times for each task.

Some of the tasks listed in the original plan turned out to be irrelevant as the project developed and moved forward. Investigations into the project's subject and the formulation of the goals and objectives also revealed additional tasks that needed to be written down in the plan. The project Gantt chart has to be updated in order to reflect this. In December 2022, a revised project Gantt chart (Appendix A) was created. A critical route was provided in this version to indicate the dependencies between tasks and when one activity must be finished before another can start. The most recent Gantt chart is seen below:

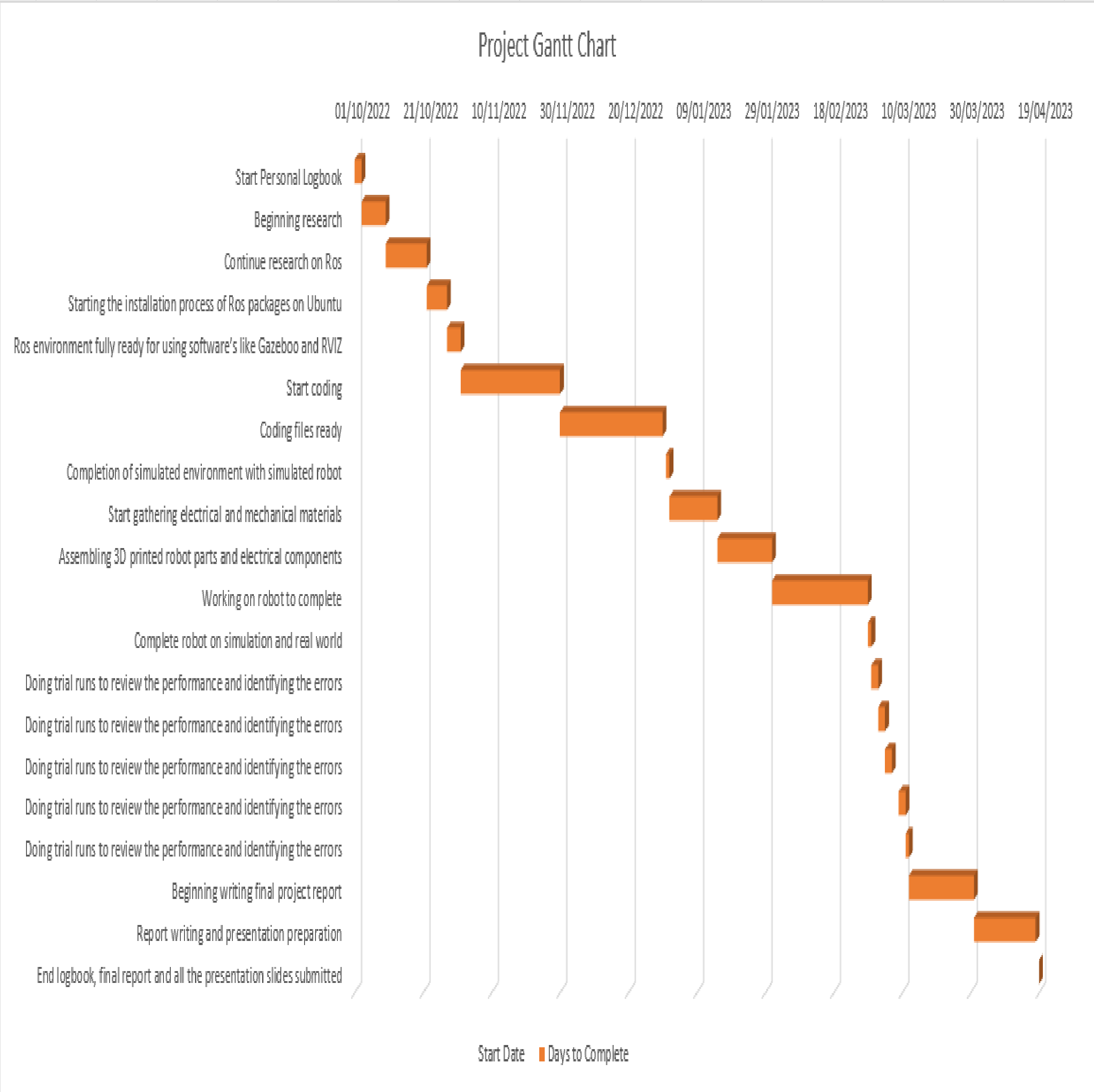


**Fig 1: Updated Project Gantt Chart.**

## 3.1 Contact Register and Logbook:

A contact register (Appendix B) was used to keep track of the time spent in contact with the project manager. In addition to this registration, a project logbook was kept. This informal journal served to keep track of project-related choices and served as a useful resource as progress was achieved. During a contact meeting with the project supervisor, if a subject that wasn't on the project Gantt Chart came up, it was noted in the logbook.

## 3.2 Risk Awareness and Mitigation:

In the interim research proposal (Appendix D), a safety risk assessment was taken into account. Due to the lack of a need for laboratory space or faculty technician time, there was no chance that these factors would be unavailable and cause a delay. The UWE engineering faculty's risk assessment was enough to cover the project's desk-based operations because, as was to be expected, it did not require any actual labour. When using a computer, proper working practices had been observed, which included positioning the chair and monitor at a comfortable height and taking frequent breaks away from the screen to prevent eye strain.

A risk assessment table was created to analyse the different types of risk events which may occur. It includes the level of the risk (Negligible<Tolerable<Moderate<Major<Fatal), the severity and likelihood of the events. It also shows the effects and mitigations of the risks. The risk assessment table is described below:

| **Risk Assessment** | | | | | |
|---|---|---|---|---|---|
| **Risks** | **Risk Level(S)** | **Risk Level(L)** | **Risk Level** | **Hazards Identified** | **Existing Control Measures** |
| Miss scheduled work | **1** | **2** | **Tolerable** | Using computer for a long period can cause health problem. | Regularly taking breaks while working. |
| Electrical failure | **1** | **1** | **Negligible** | A loose wire may result in a failed connection. | Ensuring that each wire is properly positioned. |
| Power system | **3** | **1** | **Tolerable** | While in the lab, leakage on the battery may cause fire. | Avoid using electronics for extended periods of time, and routinely check the battery's health. |
| Faulty map | **1** | **2** | **Tolerable** | Simulation robot can be misled if the map is not created properly. | Making correct map for exact place and testing it for multiple time can solve this. |
| Bug in code | **1** | **3** | **Tolerable** | Bug in the code can cause problems & the robot won't work properly. | Use clear, simple codes to reduce the amount of bug in the code. |

| | | | | | |
|---|---|---|---|---|---|
| Human errors | **2** | **3** | **Tolerable** | If experienced humans don't supervise, then may cause problems. | Working under project supervisor's supervision can help solving this problem. |
| Seating for work | **1** | **1** | **Negligible** | Posture while seated. | Setting reminders to maintain proper posture and stretching when it is possible, such as during breaks. |

## 3.3 Work Breakdown Structure:

A project is divided into smaller, easier-to-manage components using a Work Breakdown Structure (WBS), which is a hierarchical representation of the project. It's critical to structure a project, i.e., identify the many tasks and their subtasks, before beginning to plan it. A work breakdown structure was created for this project which allowed deep understanding in each of the micro works for the project.

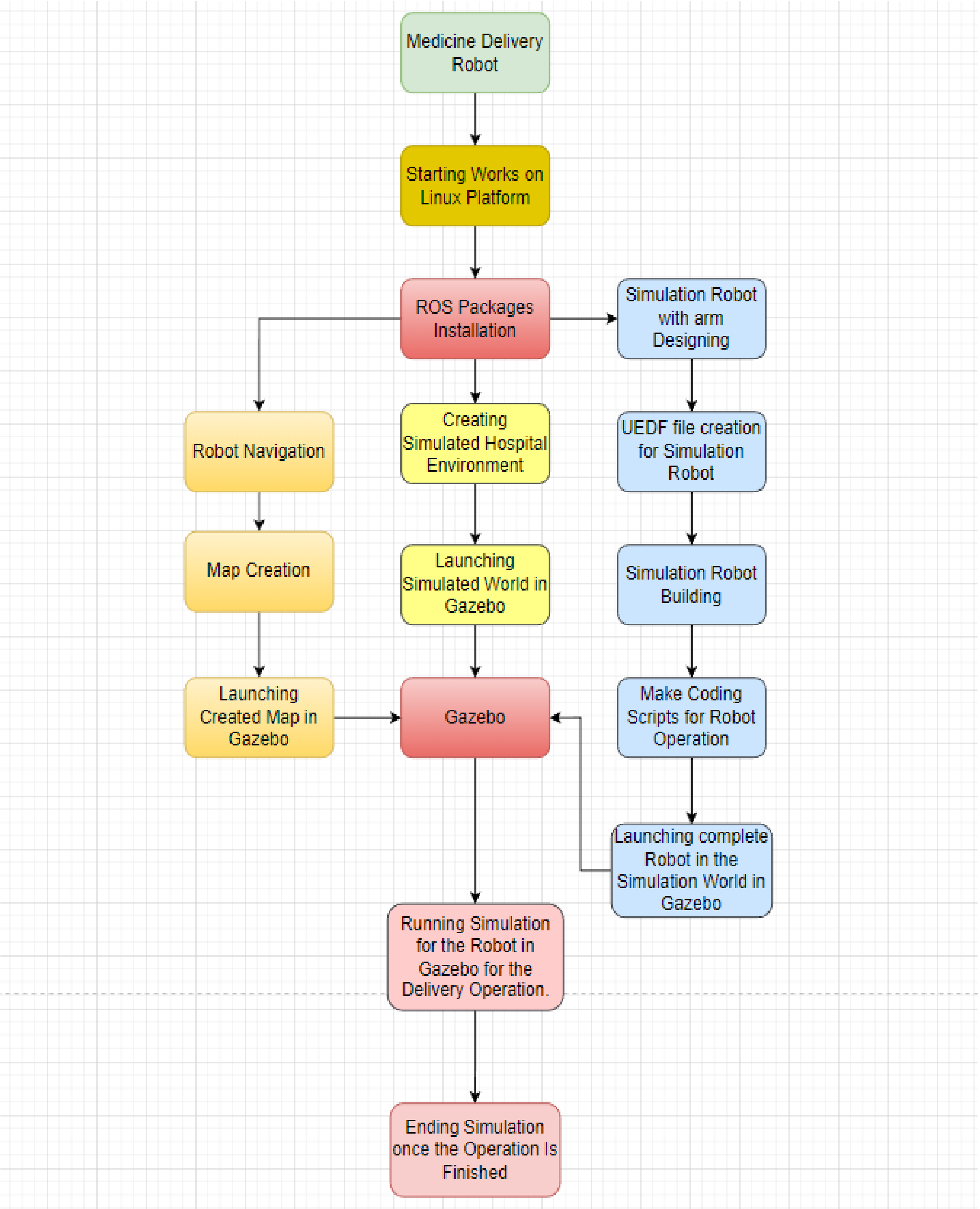


**Fig 2: Project Work Breakdown Structure.**

## 3.4 State-machine:

A state machine, also called a finite-state machine, is a computational model that is used to depict a system's behaviour. A state machine was used as a tool to model and manage the behaviour of the robot's system.

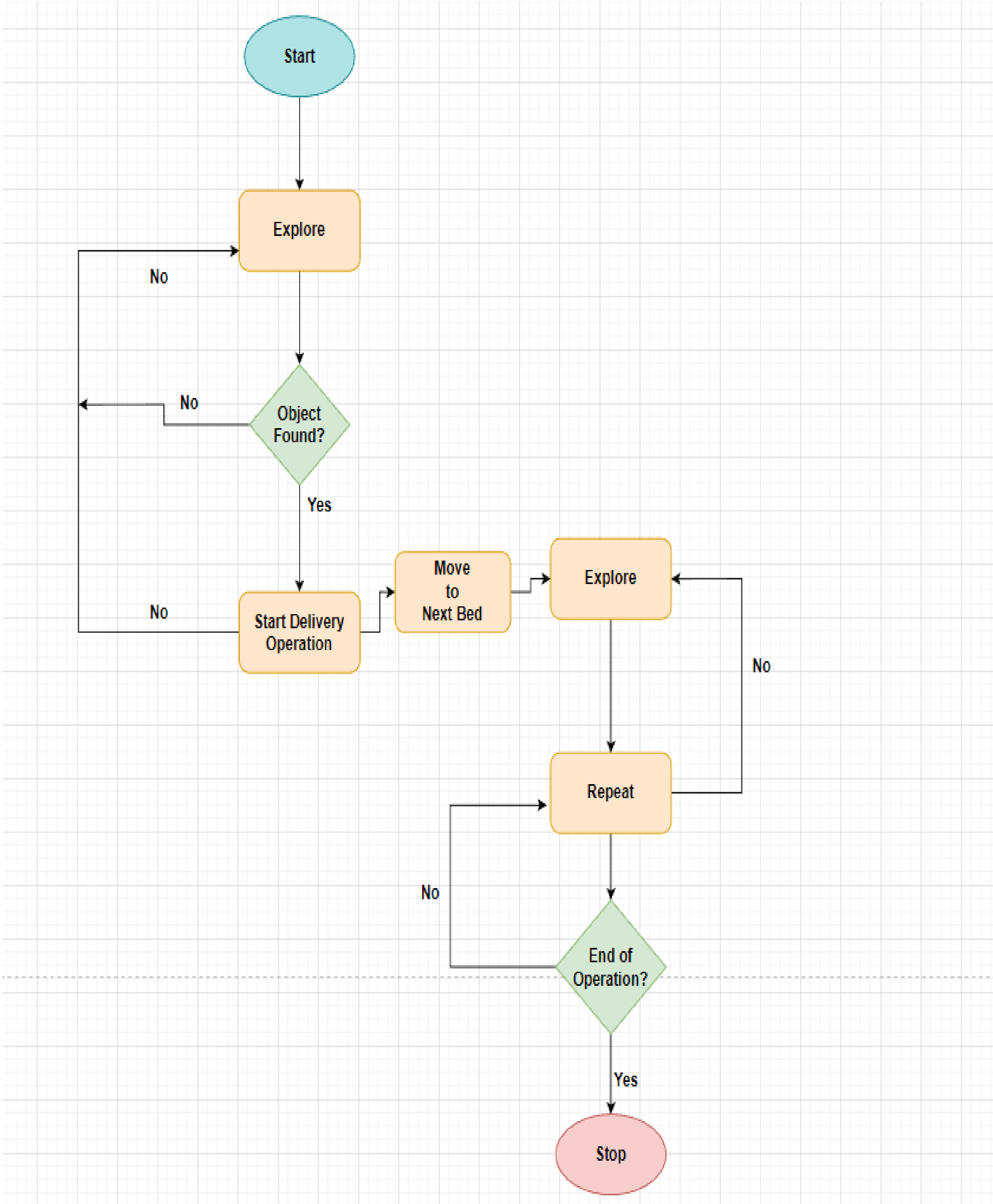


**Fig 3: Project State-machine.**

# 4. Context to the Work

Autonomous robots have gained increasing attention in recent years for their potential in delivering essential goods and services in various domains. In this context to the work section, interpretation of previous work, background research of some current autonomous delivery robots in hospital environment and critical discussion of relevant published work.

## 4.1 Background Literature Review:

For this project a simulation robot, a map from navigation and a task plan was needed. All the required literature reviews on different topics are described below:

An innovative method for creating a 3D-printed hexapod robot and modelling it in RVIZ was provided by Liu et al. (2018) in a different paper. They created the robot model in SolidWorks and visualised the robot's motion and behaviour in RVIZ. Their simulation outcomes showed how well their strategy worked for creating a working robot simulation in RVIZ.

Most recently, Lin et al. (2021) suggested a technique for creating an RVIZ ROS-based autonomous robot simulation. Their strategy entailed creating the robot model, setting up the simulation environment, and creating the communication-related ROS nodes. Their simulation outcomes showed how well their methodology worked in creating a working robot simulation in RVIZ for autonomous navigation and obstacle avoidance.

In one study, Fankhauser et al. (2016) put forth a technique for modelling the ROS navigation stack-based navigation of a mobile robot in Gazebo. They replicated the TurtleBot robot platform's navigation in a warehouse setting. Their simulation results showed how well their strategy worked in simulating actual circumstances in Gazebo.

More recently, Song et al.'s (2020) proposed a solution for modelling a drone's navigation in Gazebo. For the drone's navigation in a dynamic environment, they utilised a PID controller and a particle filter. The usefulness of their approach in replicating drone navigation in Gazebo was shown by the simulation results.

In a study by Niko et al. (2019), suggested a way for a mobile robot in Gazebo to carry out exploration and mapping tasks. Simultaneous localization and mapping (SLAM) were carried out using the ROS navigation stack and the GMapping algorithm, and the usefulness of their method was proven in Gazebo by simulating actual exploration circumstances.

In another study, Gao et al. (2020) proposed a method for a robotic arm to manipulate objects in a Gazebo. They showed the success of their methodology by replicating real-world manipulation scenarios in Gazebo using the MoveIt! package to plan and carry out the manipulation tasks.

Most recently, Li et al. (2021) suggested a method for a mobile robot to perform transportation tasks in Gazebo. They planned and carried out the transportation activities using the ROS navigation stack and the dynamic window method, demonstrating the efficiency of their strategy by simulating real-world transportation scenarios in Gazebo.

In a study by Murty et al. (2018), the effectiveness of the gmapping algorithm was assessed in environments with dynamic obstacles that move and alter over time. The study discovered that the gmapping technique can update the map in real-time and is strong enough to handle changing settings. The study also demonstrated the efficiency and computational simplicity of gmapping, demonstrating its suitability for real-time robotic applications.

In addition, a study by Huang et al. (2019) suggested a faster and more accurate variant of the gmapping technique called the FastSLAM algorithm that can create maps of large-scale environments. The study discovered that the FastSLAM approach is computationally effective for real-time applications and can handle large-scale environments with thousands of features.

## 4.2 Some Current Applications of Autonomous Robot in Hospital Environments:

The Tug autonomous medical robot from the University of California Hospital, San Francisco is a delivery robot currently working for the patients. There are two replicas available currently which can be seen walking the halls of the UCSF Medical Centre. The one that transports laundry, food, and other items resembles a pickup truck. It has a bed in the back that people move large cabinets onto and a narrower front. The second one has built-in cabinets and is boxier, more like a van which is the medicine carrier. The Tugs are not being led by any beacons. Instead, they navigate using maps stored in their minds. By using the hospital's Wi-Fi to communicate with the system, they can also detect fire alarms and move out of the way to allow carbon-based lifeforms to flee. A Tug will stop far from the elevators as it rolls down the hallways, utilising a laser and 27 infrared and ultrasonic sensors to avoid collisions, and then call an elevator down over Wi-Fi (to open doors, it uses radio waves). It will only enter an empty elevator, draw in, and then do a three-point turn to turn 180 degrees before getting off. After delivering meals to any number of floors. The robot has done this since the hospital first opened. It collects empty trays and brings them back to the kitchen, where it repeats the procedure. [ Matt Simon Science February 2015]


**Figure 4: The Tug Autonomous Medical Robot.**

Panasonic System Solutions Asia Pacific (PSSAP) and Changi General Hospital (CGH) are adopting assistive robotics technology to enhance hospital operational effectiveness as part of the nation's effort to become a Smart Nation. In phases, the Panasonic HOSPI autonomous delivery robots have been used experimentally since February 2015. The first medical facility outside of Japan to use HOSPI is CGH. The four HOSPI are able to transport large and fragile medication, medical specimens, and patient case notes around-the-clock as part of the hospital's porter management system, which relieves the load on the staff. With the help of sensors and a map of the hospital, HOSPI is trained to avoid barriers like people in wheelchairs and carry out deliveries with little assistance. It is possible to plan for new hospital routes, providing flexibility. The autonomous robot communicates with the control centre and provides information about its location, allowing its location to be tracked and always recorded. [ Panasonic Group 2015]


**Figure 5: Panasonic Autonomous Delivery Robots.**

AGV systems in hospitals**:** From the transportation to the service. AGV systems will become more important to ensure the best logistics availability and performance as hospitals start to resemble factories. Hospital AGV are taking on non-value-added duties like pushing the 800 to 1,000-pound trollies that call for an unqualified workforce as they become more and more handy and efficient. The AGV Robots for Hospitals are an important part of daily operations and assist employees in focusing on their primary areas of expertise to deliver improved patient care. AGVs can navigate by tracking tiny, embedded, cylindrical magnetic patches on the floor. Installed magnetic dots are spaced every 250–500 mm (almost 15 ft), forming a fictitious grid. The AGVs use sensors and controllers, including hall-effect sensors, counters, gyro sensors, and other types of encoders, to calibrate against steering angle faults as they move from one location to the next. It is extremely invasive to install. The area of the floor where the magnet will be placed needs to have a small hole made in it. After that, epoxy glue is used to cover the hole. [ AGV in Hospitals. Autonomous Mobile Robots Disrupting Healthcare Automation. 2020]

**Figure 6: WDR01C Model.**

## 4.3 Interpretation of Previous Work in the Topic:

To create a conceptual framework and design for the robot, the interpretation of earlier work on the subject entails studying and synthesising the current literature. The interpretation should assess the prior work critically to determine the flaws and restrictions in the current strategies and suggest a creative remedy that can handle the problems found.

Most of the academically published research on autonomous delivery robots in a hospital setting has concentrated on the robots' technical features, navigational capabilities, and human relations. The robots in this research have been given the ability to move safely across dynamic hospital environments while avoiding hazards and engaging with people by using a range of sensors, algorithms, and control techniques.

One study examined how autonomous delivery robots could navigate hospital hallways and avoid obstacles including patients, guests, and other equipment by using vision-based algorithms. This work was published in the Journal of Intelligent & Robotic Systems. According to the study, using vision-based algorithms made it easier for the robot to recognise and avoid obstacles, making its navigation system more effective and secure. (Lee J, Lee H. and Kim J. Vision-Based Obstacle Detection and Avoidance for Autonomous Delivery Robots in a Hospital Corridor Environment. 2017)

Another study, published in the Journal of Healthcare Engineering, focused on the design and development of an autonomous delivery robot that could transport medical supplies and equipment in a hospital environment. The study proposed a modular design that allowed for the customization of the robot's payload based on specific hospital needs, such as medication delivery or laboratory sample transportation. (Lee S. and Choi H. Design and Implementation of a Modular Autonomous Delivery Robot for Healthcare Services. 2018)

In the field of commercial research, many businesses have developed self-governing delivery robots that are especially designed for use in medical facilities. The TUG robot from Aethon, which can deliver supplies and equipment on its own to various areas across the hospital, is one of the most well-known options for hospital logistics. Relay, a custom autonomous robot created by Savioke, excels in delivering medical supplies and other essential goods to patients inside their own rooms. (Aethon. Tug Autonomous Mobile Robot and Svioke. Relay Autonomous Delivery Robot. 2018).

Before starting work on the development of Bumblebot pre-built virtual environment from internet was experimented. The experiment robot was turtlebot and the robot model was waffle pi. With the help of project supervisor, the robot was launched into the pre-built virtual environment. A readymade path was also tested with the turtlebot to see how it navigates through different obstacles.

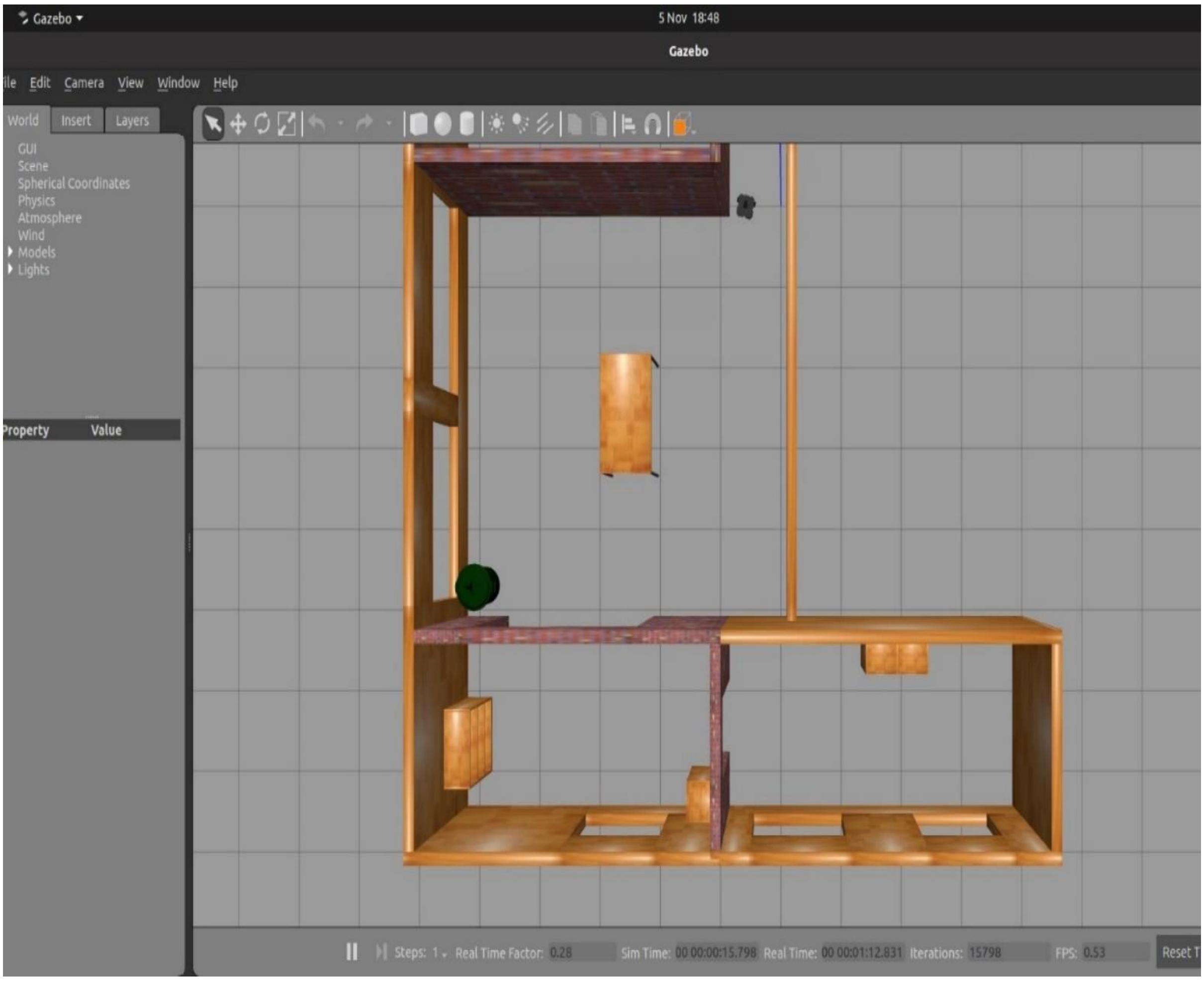


**Figure 7: Turtlebot in research virtual environment.**

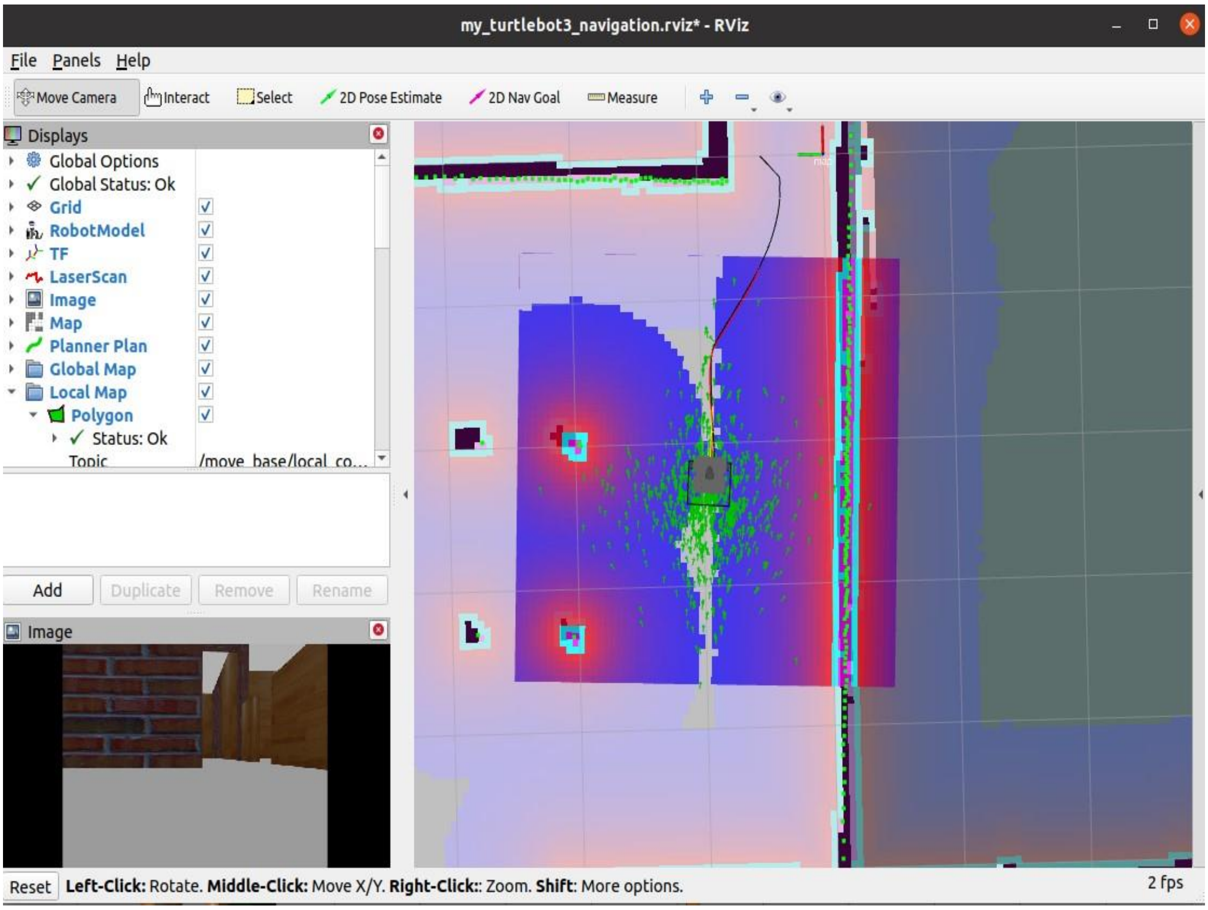


**Figure 8: Turtlebot started navigation.**

## 4.4 Critical Discussion of Relevant Published Work:

The creation of autonomous delivery robots for healthcare environments had been the subject of extensive academic and industrial research. There were still a number of issues that needed to be resolved, despite the fact that this research has shown how these robots have the potential to increase efficiency and safety in healthcare logistics.

Assuring the safety and dependability of autonomous delivery robots in hospital environment was one of the major issues they face. These robots must be able to move safely through the dynamic, unpredictable settings of hospitals while avoiding hazards like people, objects, and furniture. There was still a need for more research on how to make sure these algorithms are accurate and dependable in all circumstances, even though several studies have presented vision-based algorithms to enhance the robots' capability of obstacle identification and avoidance.

In medical environment, autonomous delivery robots' interactions with people provided another difficulty. These robots must be capable of engaging in acceptable social distancing and avoidance of contact while interacting with patients, guests, and medical employees. There was still a need for more research on how to make sure these interactions were successful and socially acceptable. Some studies had suggested employing speech recognition and natural language processing to enable these robots to converse with humans.

Additionally, there were ethical questions raised by the creation and use of autonomous delivery robots in hospital settings. For instance, there is a chance that these robots will replace human labour, which could result in job losses and exacerbate already-existing social inequities. To guarantee that these robots were utilised in healthcare settings safely and ethically, there was also a need for defined laws and standards.

Overall, there were still a number of issues that need to be resolved despite the published research's ability to show how autonomous delivery robots in hospital environments could improve healthcare logistics. Future studies should concentrate on enhancing the reliability and safety of the robots, creating efficient and socially acceptable human interactions, and resolving any ethical issues that may arise.

# 5. Research Methodology

Tools like Gazebo and SweetHome3D was used to create a simulated environment. The behaviour and performance of the physical robot was tested and evaluated in a secure and controlled environment by using a simulation robot to mimic the physical robot in the virtual environment. The simulation robot was given robot commands through python script to carry out activities like moving to various areas and dispensing medication to patients. For the robot to explore the simulated environment with accuracy, a navigation map was built utilising methods like SLAM (Simultaneous Localization and Mapping). In order to launch all the robot tasks different launch files were made. Together, these components can be utilised to test and assess the robot's performance and behaviour prior to its introduction into a real hospital setting.

## 5.1 Requirement Analysis:

The requirements for designing and building the Bumblebot for medicine delivery in hospital environment sectioned into two parts. They are user requirements, mechanical requirements and system requirements.

**User Requirements:**

1. Safety: The robot must operate in a safe manner while inside the hospital, including avoiding collisions with tools.
2. Accuracy: The robot must give medication to the right patient at the right time and place.
3. Flexibility: The robot must be able to adjust to various hospital surroundings and layouts (e.g., different wards), including avoiding obstructions.
4. Reliability: The robot must be reliable and consistent in its performance, ensuring timely and efficient delivery of medications.
5. User-friendly interface: The hospital personnel should be able to easily monitor the robot's progress and receive notifications if something is wrong thanks to its user-friendly interface.
6. Minimal disruption: The robot's operation should be silent and unobtrusive to cause the least amount of inconvenience to patients and employees.
7. Performance Vision: Live footage screen of simulated world for operator on Gazebo and Rviz.

**Mechanical Requirements:**

1. Mobility: The robot should be able to move freely throughout the facility, including through doors and around corners.
2. Payload: The robot should be able to transport a variety of medicine payloads of various weights and sizes.
3. Operation distance: The robot should have a service distance of 23 meters for three words in one go.
4. Battery life: Battery life of the robot running system the computer in this case must have long lasting battery life.

**System Requirements:**

1. Navigation: The robot should be equipped with a precise and trustworthy navigation system that can work in a hospital setting, including avoiding obstructions.
2. Localization: The robot should be able to accurately determine its position in the hospital environment.

3. Automation: The robot must be able to function autonomously, which includes recognising and reacting to changes in the hospital environment.
4. Sensor suite: The sensors carried by the robot, should provide accurate signal to help it navigate and avoid obstacles.
5. Robustness: To ensure that the robot can recover from mistakes or unforeseen circumstances, the software should be developed to be fault-tolerant and robust.

## 5.2 Virtual Environment Design:

The process normally began with the identification of the project's main goals and constraints, such as the design and dimensions of each ward. Once these specifications are known, a concept sketch of the hospital environment having three wards and 15 beds in total was made. There were also given door and windows in each ward inside the concept sketch. Before creating the hospital environment in SweetHome3D website the concept sketch made a crucial impact in finishing the virtual environment successfully.

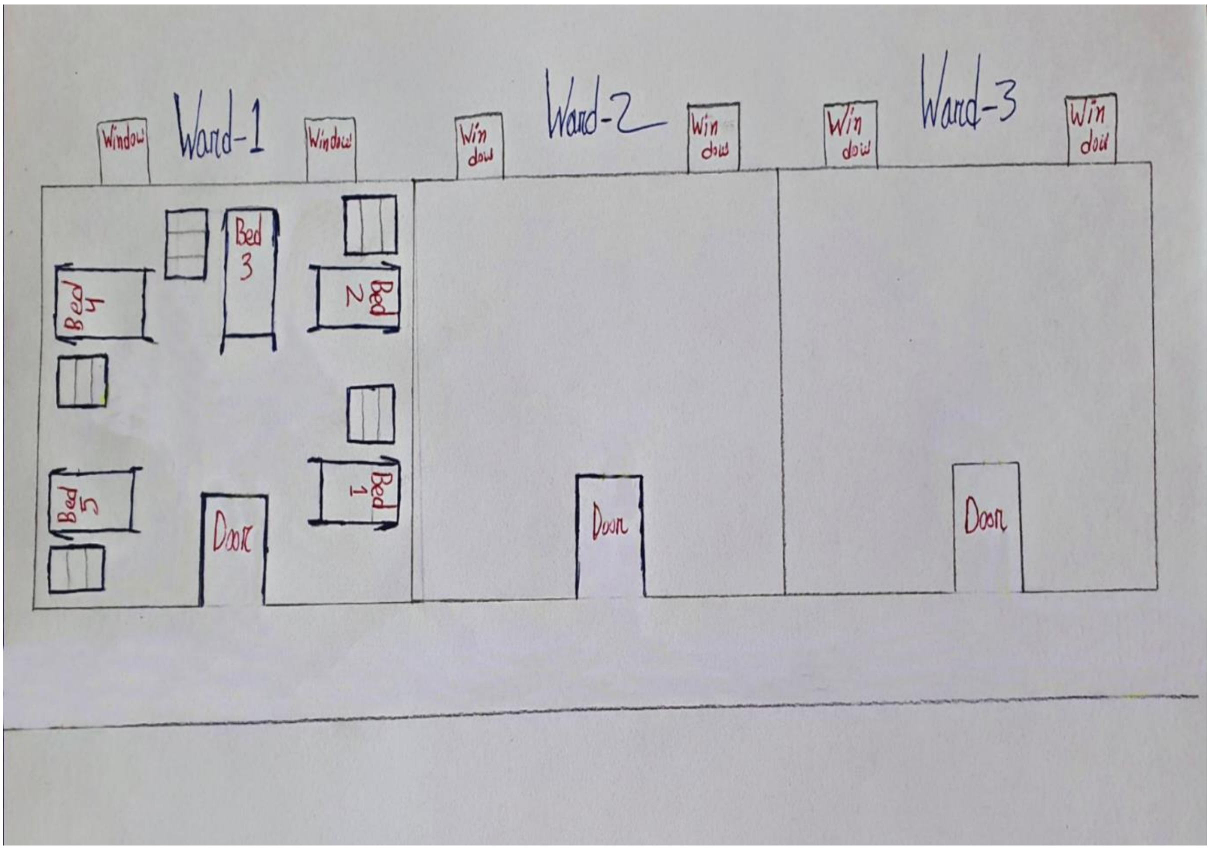


**Figure 9: Concept Hospital Environment Sketch.**

After getting the concept sketch an idea was generated on how the design of the virtual environment had to be. Next it was searching for a software or website where a virtual environment could have been created. To create 3D models of buildings and interiors, including hospitals, using the free and open-source interior design programme SweetHome3D was found from the research. Although there were 5 beds in each ward on the initial plan, later it was changed to 2 beds per wards in order to decrease complexity on robot’s navigation. The number of wards were stayed same as planed which was 3. The created environment was saved as a .obj format and later imported in Gazebo for robot’s virtual world. The final completed virtual ward environment is shown below:

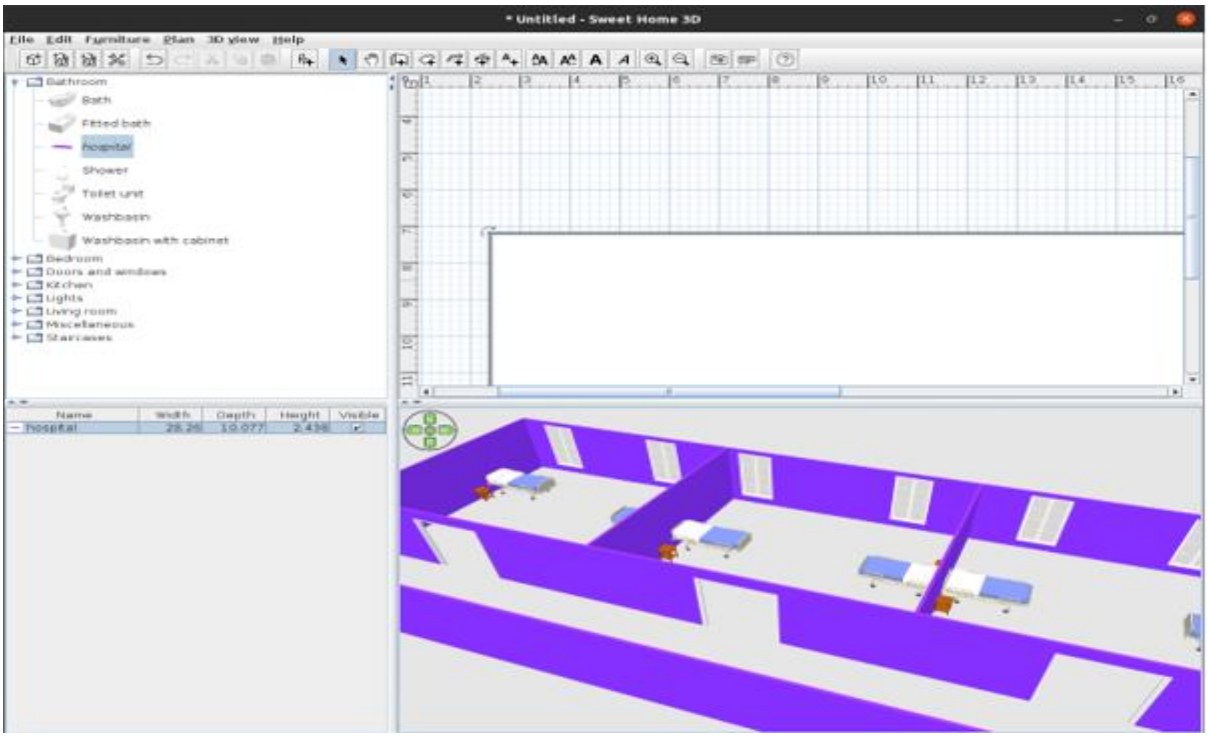

**Fig 10: Created 3D Hospital Environment in SweetHome3D**

## 5.3 Bumblebot Robot with its URDF

For this Bumblebot project a new robot was planned to make. The inspiration came from several online models such as turtlebot. Main goal for the robot creation was to create something easy and simple shaped so that it can be completed by the timeframe reserved for robot creation. From different types of research, a cylindrical shape came under project maker’s consideration and later proceeded with the shape. The robot was planned to make with 2 main wheels connected with motor and 2 castor wheels attached in front and rear of the robot. The basket to carry medicines will be placed with the robot body. On top part of the robot the lidar and an arm will be placed. A concept sketch was drawn before start designing the original robot.

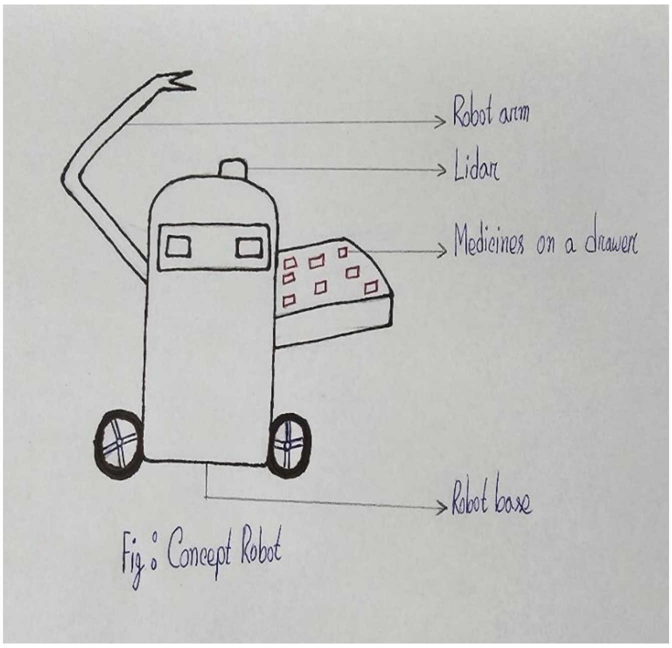


**Figure 11: Concept Robot.**

For the creation of Bumblebot a URDF (Unified Robot Description Format) file was a must to make. An explanation about URDF is given below:

An XML-based file format called URDF (Unified Robot Description Format) is used to define a robot's physical characteristics for usage in simulations, visualisation, and other applications. Typically, a URDF file contains details on the robot's kinematic structure, joint characteristics, visual and collision meshes, as well as physical characteristics like mass and inertia. Roboticists can build intricate robot models using URDF files because they have a standardised format that is simple to distribute and utilise in simulation settings. This makes it simpler to simulate and test robotic systems before putting them into use in the real world.

The main components of a URDF file for a virtual robot are:
**Links:** The physical components of the robot, such as its body, arms, wheels and other parts.
**Joints:** The connections between the links the allow the robot to move.
**Sensors**: Information about sensors on the robot, such as cameras, lidars or torque sensors.
**Visual and Collision Geometry:** The robot's visual representation, which may include 3D renderings of its parts, and the collision detection geometry.

All the links and joints for Bumblebot were planned before start working on the URDF file. The links and joints of Bumblebot are picturised below:

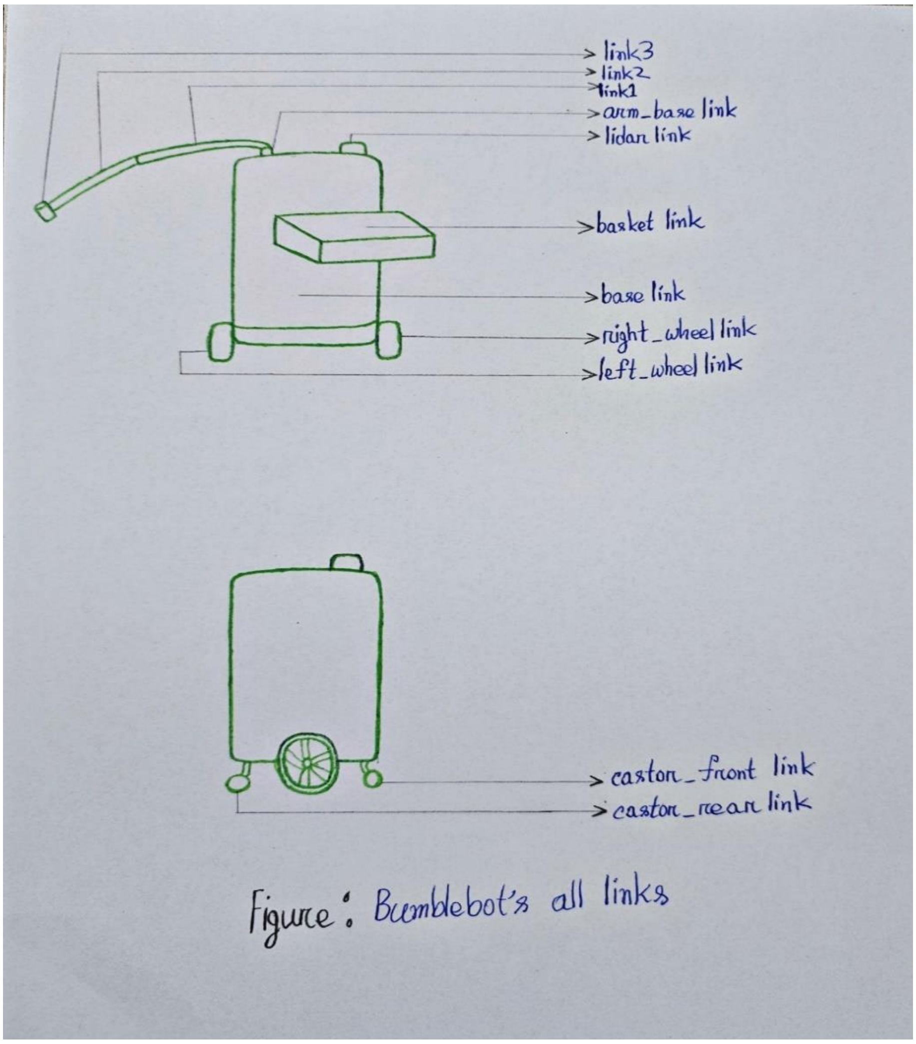


**Figure 12: Bumblebot’s all links.**

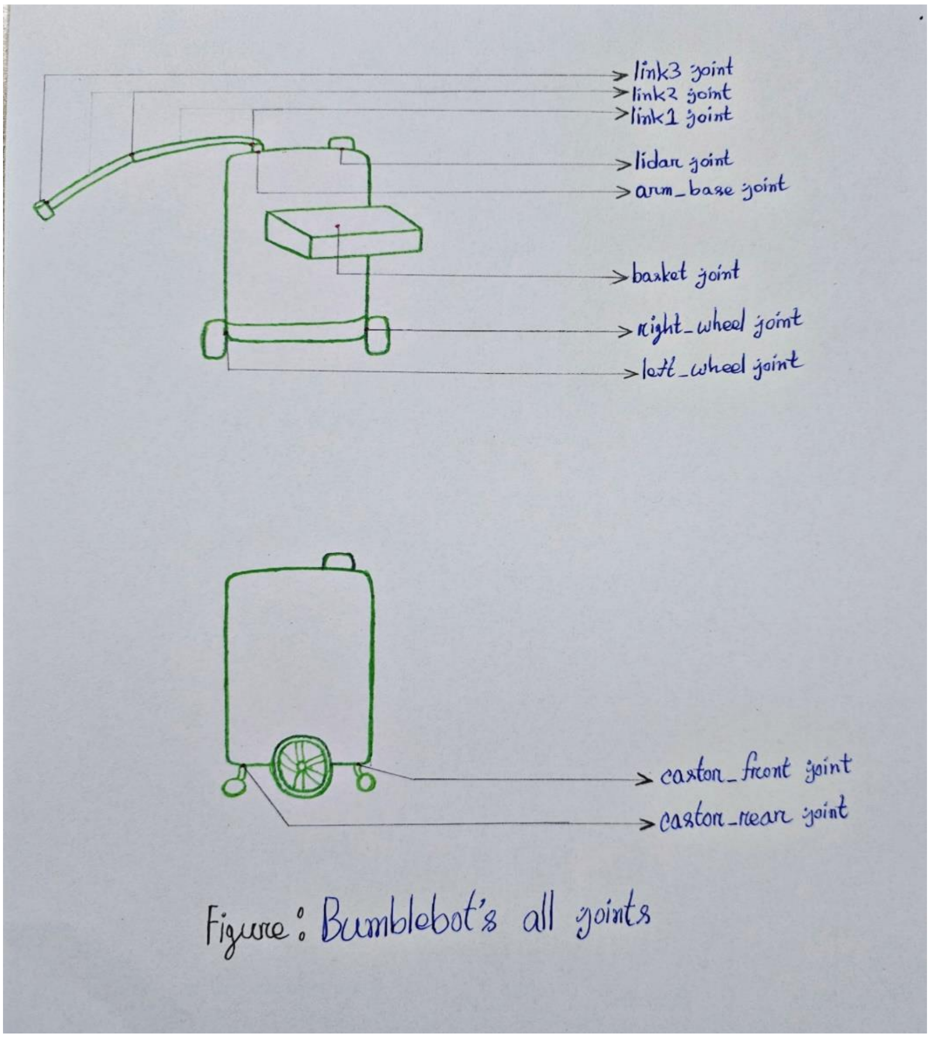


**Figure 13: Bumblebot's all joints.**

For the Bumblebot's links and joints in the URDF, all the identified links and joints are given below:

**base** link:

```
<link name="base_link">
  <inertial>
    <mass value="100.0"/>
<inertia  ixx="3.416666706403096" ixy="0" ixz="0" iyy="3.416666706403096" iyz="0"
izz="2.6666667461395264" />
  </inertial>
  <visual>
    <origin
      xyz="0 0 0.25"
      rpy="0 0 0" />
    <geometry>
      <cylinder radius="0.2" length="0.5"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0.25"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.2" length="0.5"/>
    </geometry>
  </collision>
</link>
```

**left_wheel** link and joint**:** The joint for this one is continuous.

```
<gazebo reference="base_link">
  <material>Gazebo/Yellow</material>
</gazebo>

<link name="left_wheel">
  <inertial>
    <mass value="1"/>
    <inertia ixx="0.01" iyy="0.01" izz="0.01"
    ixy="0" ixz="0" iyz="0"/>
  </inertial>
    <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.07" length="0.06"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.07" length="0.06"/>
    </geometry>
  </collision>
</link>

<gazebo reference="left_wheel">
  <material>Gazebo/Black</material>
</gazebo>

  <joint
    name="left_wheel_joint"
    type="continuous">
    <origin
        xyz="0 0.2 0.065"
        rpy="-1.57 0 0" />
    <parent
      link="base_link" />
    <child
      link="left_wheel" />
    <axis
      xyz="0 0 1" />
  </joint>


```

**Figure 14: left_wheel link and joint in URDF.**

**right_wheel** link and joint**:** The joint for this one is continuous.

```
<link name="right_wheel">
  <inertial>
    <mass value="1"/>
    <inertia ixx="0.01" iyy="0.01" izz="0.01"
    ixy="0" ixz="0" iyz="0"/>
  </inertial>
  <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.07" length="0.06"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.07" length="0.06"/>
    </geometry>
  </collision>
</link>


<gazebo reference="right_wheel">
  <material>Gazebo/Black</material>
</gazebo>

  <joint
    name="right_wheel_joint"
    type="continuous">
    <origin
        xyz="0 -0.2 0.065"
        rpy="-1.57 0 0" />
    <parent
      link="base_link" />
    <child
      link="right_wheel" />
    <axis
      xyz="0 0 1" />
  </joint>
```

**Figure 15: right_wheel link and joint in URDF.**

**caster_front** link and joint**:** The joint for this one is fixed.

```xml
<link name="caster_front">
  <inertial>
    <mass value="0.3"/>
    <inertia ixx="0.01" iyy="0.01" izz="0.01"
    ixy="0" ixz="0" iyz="0"/>
  </inertial>
  <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <sphere radius="0.03"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <sphere radius="0.03"/>
    </geometry>
  </collision>
</link>

  <joint
    name="caster_front_joint"
    type="fixed">
    <origin
        xyz="0.17 0 0.025"
        rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="caster_front" />
    <axis
      xyz="0 1 0" />
  </joint>

```

**Figure 16: caster_front link and joint in URDF.**

**caster_back** link and joint: The joint for this one is fixed.

```
<link name="caster_back">
  <inertial>
    <mass value="0.3"/>
    <inertia ixx="0.01" iyy="0.01" izz="0.01"
    ixy="0" ixz="0" iyz="0"/>
  </inertial>
  <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <sphere radius="0.03"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <sphere radius="0.03"/>
    </geometry>
  </collision>
</link>

  <joint
    name="caster_back_joint"
    type="fixed">
    <origin
        xyz="-0.17 0 0.025"
        rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="caster_back" />
    <axis
      xyz="0 1 0" />
  </joint>

```

**Figure 17: caster_back link and joint in URDF.**

**lidar** link and joint: The joint for this one is fixed.

```
<link name="lidar">
  <inertial>
    <mass value="0.2"/>
    <inertia ixx="0.001" iyy="0.001" izz="0.001"
    ixy="0" ixz="0" iyz="0"/>
  </inertial>
  <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.025" length="0.035"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.025" length="0.04"/>
    </geometry>
  </collision>
</link>

  <joint
    name="lidar"
    type="fixed">
    <origin
        xyz="0.15 0 0.517"
        rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="lidar" />
  </joint>

```

**Figure 18: lidar link and joint in URDF.**

**basket** link and joint: The joint for this one is fixed.

```
<link name="basket">
  <inertial>
    <mass value="0.5"/>
<inertia  ixx="0.0018254168158769641" ixy="0" ixz="0" iyy="0.0018254168158769641"
iyz="0" izz="0.0034340836398601597" />
  </inertial>
  <visual>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <mesh filename="package://hospital/meshes/basket.stl"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0"
      rpy="0 0 0" />
    <geometry>
    <mesh filename="package://hospital/meshes/basket_collsion.stl"/>
    </geometry>
  </collision>
</link>

<gazebo reference="basket">
  <material>Gazebo/Green</material>
</gazebo>

  <joint
    name="basket_joint"
    type="fixed">
    <origin
        xyz="-0.2 0 0.474"
        rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="basket" />
  </joint>

```

**Figure 19: basket link and joint in URDF.**

**arm_base** link and joint: The joint for this one is revolute.

```
<link name="arm_base_link">
  <inertial>
    <mass value="0.1"/>
<inertia  ixx="0.01" ixy="0" ixz="0" iyy="0.01" iyz="0" izz="0.01" />
  </inertial>
  <visual>
    <origin
      xyz="0 0 0.025"
      rpy="0 0 0" />
    <geometry>
      <cylinder radius="0.015" length="0.051"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0.025"
      rpy="0 0 0" />
    <geometry>
    <cylinder radius="0.015" length="0.05"/>
    </geometry>
  </collision>
</link>

<gazebo reference="arm_base_link">
  <material>Gazebo/Black</material>
</gazebo>

  <joint
    name="arm_base_link_joint"
    type="revolute">
    <origin
        xyz="-0.1 0 0.5"
        rpy="0 0 0" />
    <parent
      link="base_link" />
    <child
      link="arm_base_link" />
      <limit effort="1000.0" lower="0" upper="3.14" velocity="0.5"/>
    <axis
      xyz="0 0 1" />
  </joint>
```

**Figure 20: arm_base link and joint in URDF.**

**link1 or 1st part of the robot arm's** link and joint: The joint for this one is fixed.

```
<link name="link_1">
  <inertial>
    <mass value="0.05"/>
<inertia  ixx="0.01" ixy="0" ixz="0" iyy="0.01" iyz="0" izz="0.01" />
  </inertial>
  <visual>
    <origin
      xyz="0 0.15 0.015"
      rpy="0 0 0" />
    <geometry>
      <box size="0.03 0.3 0.03"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0.15 0.015"
      rpy="0 0 0" />
    <geometry>
    <box size="0.03 0.3 0.03"/>
    </geometry>
  </collision>
</link>

<gazebo reference="link_1">
  <material>Gazebo/Red</material>
</gazebo>

  <joint
    name="link_1_joint"
    type="fixed">
    <origin
        xyz="0 0 0.02"
        rpy="0 0 0" />
    <parent
      link="arm_base_link" />
    <child
      link="link_1" />
  </joint>

```

**Figure 21: link1's link and joint in URDF.**

**link2 or 2nd part of the arm's** link and joint: The joint for this one is revolute.

```xml
<link name="link_2_base">
  <inertial>
    <mass value="0.05"/>
<inertia  ixx="0.01" ixy="0" ixz="0" iyy="0.01" iyz="0" izz="0.01" />
  </inertial>
  <visual>
    <origin
      xyz="0 0 0.015"
      rpy="0 0 0" />
    <geometry>
       <cylinder radius="0.015" length="0.031"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0.015"
      rpy="0 0 0" />
    <geometry>
     <cylinder radius="0.015" length="0.03"/>
    </geometry>
  </collision>
</link>

<gazebo reference="link_2_base">
  <material>Gazebo/Black</material>
</gazebo>

 <joint
   name="link_2_base_joint"
   type="revolute">
   <origin
       xyz="0 0.3 0.0"
       rpy="0 0 0" />
   <limit effort="1000.0" lower="0" upper="3.14" velocity="0.5"/>
   <parent
     link="link_1" />
   <child
     link="link_2_base" />
   <axis
     xyz="0 0 1" />
 </joint>
```

**Figure 22: link2's link and joint in URDF.**

**link3 or end effector of the robot arm's** link and joint: The joint for this one is continuous.

```
<link name="link_3">
  <inertial>
    <mass value="0.01"/>
<inertia  ixx="0.01" ixy="0" ixz="0" iyy="0.01" iyz="0" izz="0.01" />
  </inertial>
  <visual>
    <origin
      xyz="0 0 0.032"
      rpy="0 0 0" />
    <geometry>
       <cylinder radius="0.015" length="0.05"/>
    </geometry>
  </visual>
  <collision>
    <origin
      xyz="0 0 0.032"
      rpy="0 0 0" />
    <geometry>
     <cylinder radius="0.015" length="0.05"/>
    </geometry>
  </collision>
</link>

<gazebo reference="link_3">
  <material>Gazebo/Black</material>
</gazebo>

  <joint
    name="link_3_joint"
    type="continuous">
    <origin
        xyz="0 0.3 -0.015"
        rpy="0 0 0" />
    <parent
      link="link_2" />
    <child
      link="link_3" />
    <axis
      xyz="0 0 1" />
  </joint>
```

**Figure 23: link3's link and joint in URDF.**

## 5.4 Robot Navigation and Localization:

Robot’s navigation started with a hand drawn path plan. At the initial stage of the project a hand drawn path was planned.

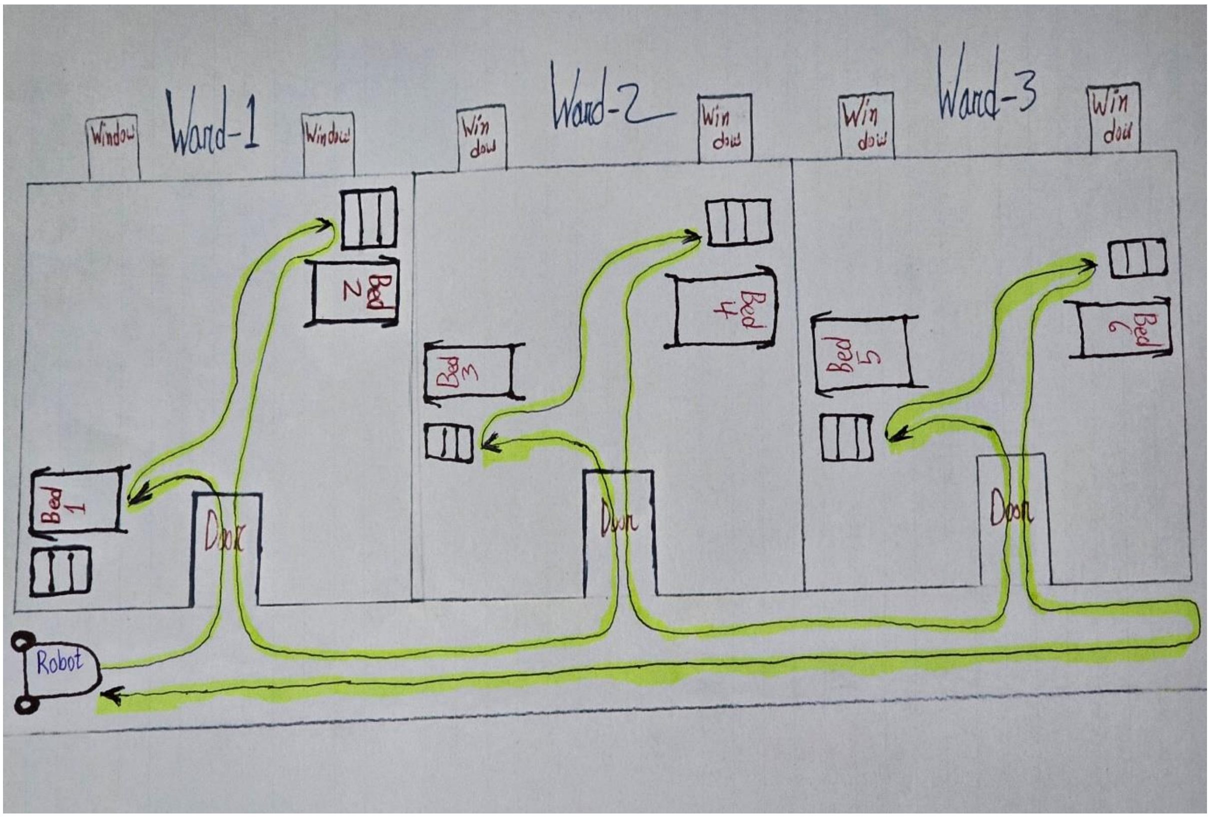


**Figure 24: Hand drawn path plan.**

SLAM gmapping package of Ros was used for mapping. Rao-Blackwellized particle filters have recently been developed as a successful method for resolving the SLAM (simultaneous localization and mapping) issue. This method makes use of a particle filter where each particle carries its own map of the surrounding area. Therefore, a crucial query was how to lower the particle count. For learning grid maps, it was described adaptive strategies to lower the particle count in a Rao-Blackwellized particle filter. Gmapping suggested a method to compute an accurate suggestion distribution that considers both the most recent observation and the robot's movement. This significantly reduces the filter's prediction step's uncertainty regarding the robot's pose. Additionally, Gmapping employed a method for performing selective resampling operations that significantly lessens the issue of particle depletion.

In the Bumblebot project, a map of the environment was made using the SLAM (Simultaneous Localization and Mapping) gmapping technique, and the robot was then located within it. A laser range finder was used by the ROS module Gmapping to map the surrounding area and pinpoint the location of the robot within it. It used information from sensors (lidar) to map the environment and determine the robot's location. A two-dimensional occupancy grid map was produced using data processed from the laser range finder's scan of the surrounding area. Based on information from the robot's sensors, this map was then utilised to infer the robot's location within the surrounding space. As the robot moved through the environment, SLAM gmapping was used to update the map and the robot's position in real time. As a result, the robot could move around its surroundings without the aid of outside sensors or pre-made maps. The staring state of Bumblebot for navigation using Gmapping package is shown below:

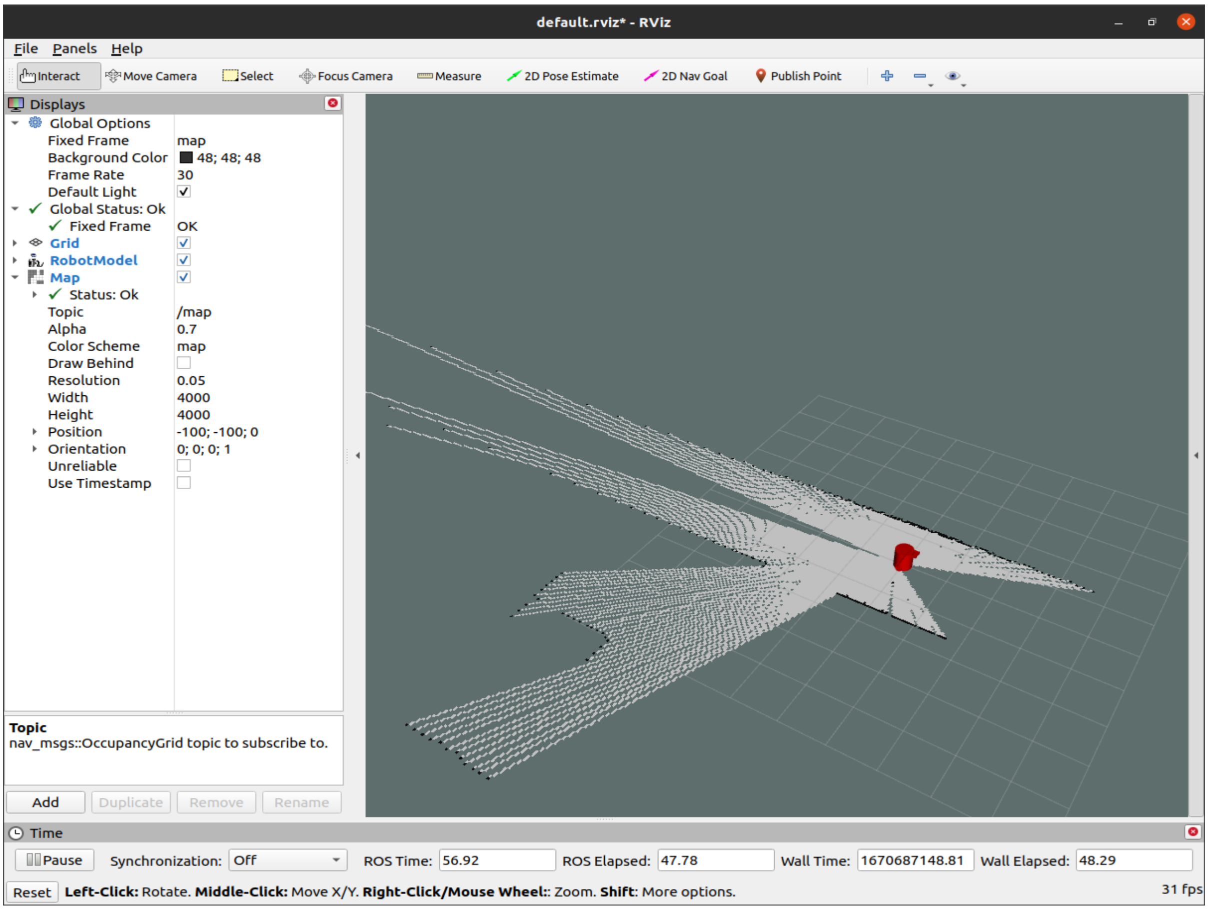


**Figure 25: Robot has started exploring by using Gmaping.**

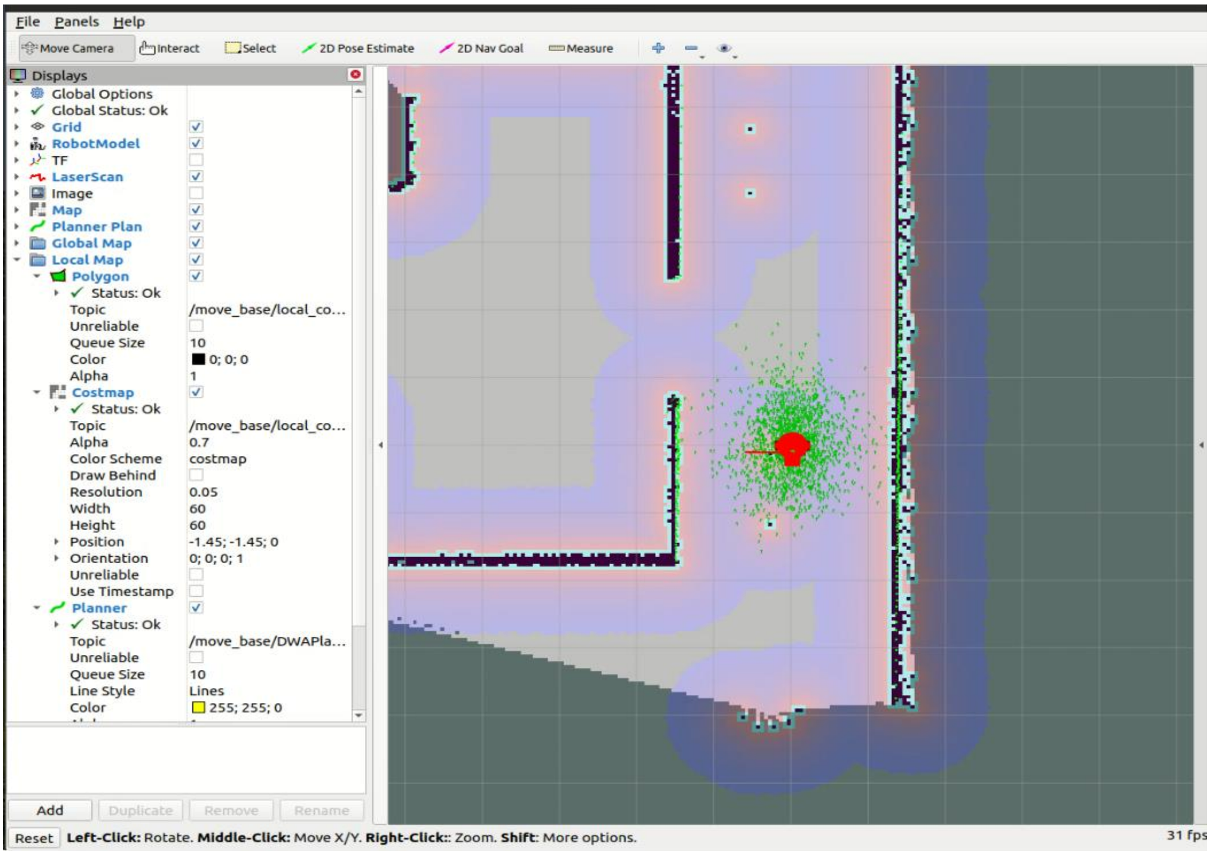


**Figure 26: Robot is about to start navigation.**

## 5.5 Bumblebot's Task Planner

Before writing the python coding script, a flowchart was made for having deep understanding for the code. After that the python script was written where a sequence of different tasks was programmed. The Bumblebot first puts all the medicine inside the basket and closes the arm and then navigates to the table where it picks up medicine and drops it on the side of the hospital bed. The same operation procedure repeats for the rest of the wards.

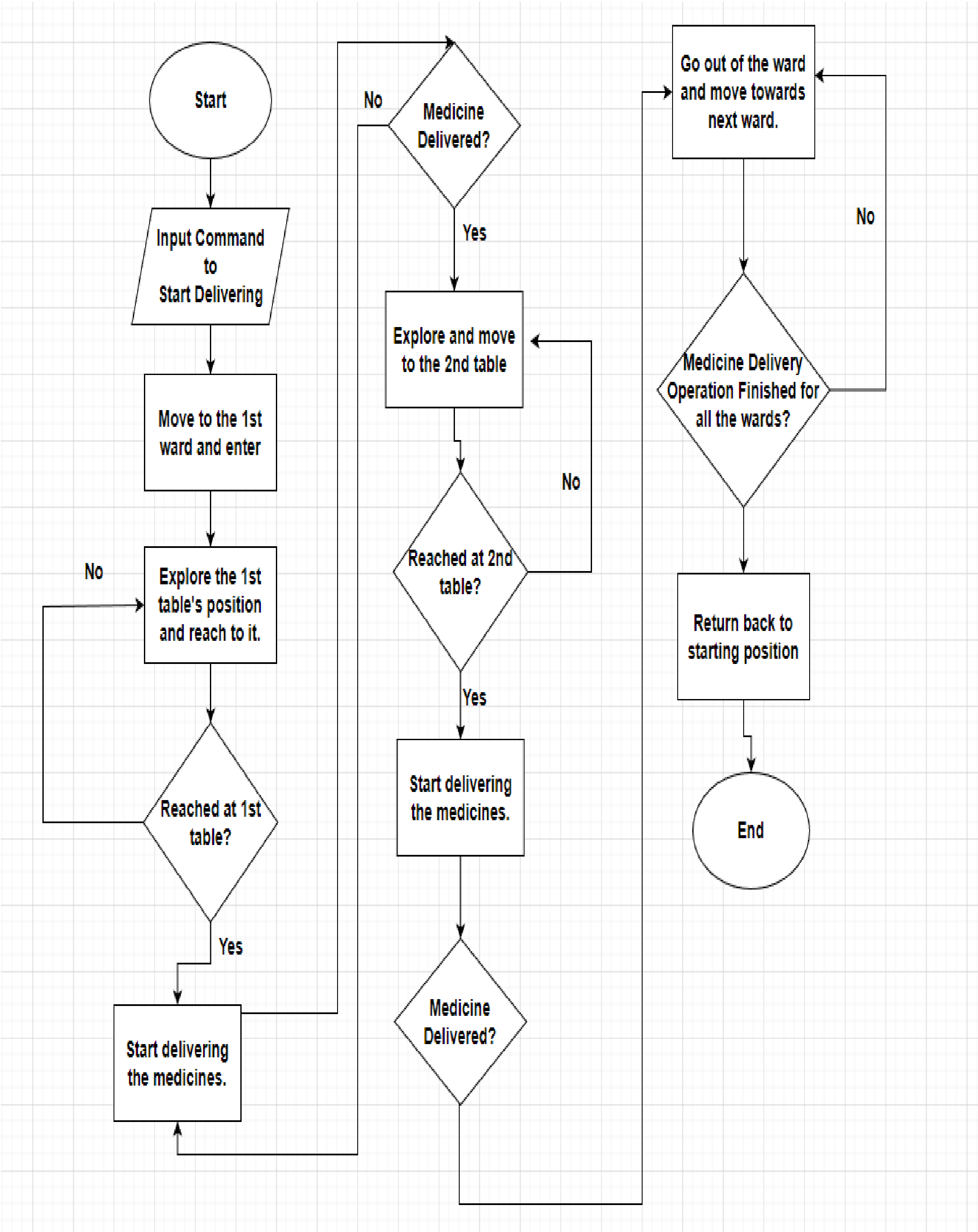


**Figure 27: Flowchart for robot task programming.**

During the initial stage of coding all the necessary packages were added for further works.

```python
#!/usr/bin/env python3
import rospy
from std_msgs.msg import Float64
import actionlib
import tf
from gazebo_msgs.msg import ModelState
from gazebo_msgs.srv import SetModelState, GetLinkState
from move_base_msgs.msg import MoveBaseAction, MoveBaseGoal

```

**Fig 28: Necessary packages**

All the necessary packages are explained below:

**rospy**: This package offers the fundamental capabilities for building ROS nodes in Python. By publishing and subscribing to topics, it enables the code to interact with other nodes.

**std_msgs.msg**: This package offers common message kinds for ROS. The Float64 message type, which is used to convey floating point values, is being imported into the code in this instance.

**actionlib:** To create and manage asynchronous jobs in ROS, this module was used. For time-consuming operations like navigation, it enables the code to transmit and receive feedback and status updates.

**tf:** This package offered ROS user the tools needed to work with coordinate frame transforms. It enabled the code to apply transformations and switch between various coordinate frames.

**gazebo_msgs.msg:** This package offers the message types required to communicate with the Gazebo simulator. In this instance, the code imports the **'ModelState**' message type, which was used to control model robot's pose.

**gazebo_msgs.srv**: For interfacing with the Gazebo simulator, this package offers many service types. In this instance, the code is importing the 'SetModelState' and 'GetLinkState' service types, used for establishing the robot model's posture and retrieving a link's state in the Gazebo model, respectively.

**move_base_msgs.msg**: For directing a robot's navigation, this package offered message types. In this instance, the code was importing the message types 'MoveBaseAction' and 'MoveBaseGoal', which were used to convey navigation goals to the robot's move base.

The robot task functions are explained below:

Inside the main function the tasks are coded with appropriate comment for the reader to understand. A goal function was created for the robot for its pose and orientation in every goal.

```
def goal(a,b,c):
    goal = MoveBaseGoal()
    goal.target_pose.header.frame_id = "odom"
    goal.target_pose.header.stamp = rospy.Time.now()
    goal.target_pose.pose.position.x = a
    goal.target_pose.pose.position.y = b
    quaternion = tf.transformations.quaternion_from_euler(0.0, 0.0, c)
    goal.target_pose.pose.orientation.x = quaternion[0]
    goal.target_pose.pose.orientation.y = quaternion[1]
    goal.target_pose.pose.orientation.z = quaternion[2]
    goal.target_pose.pose.orientation.w = quaternion[3]

    client.send_goal(goal)
    wait = client.wait_for_result()
    if not wait:
        rospy.logerr("Action server not available!")
        rospy.signal_shutdown("Action server not available!")
    else:
        return client.get_result()

```

**Figure 29: goal function**

The robot starts form its initial position outside the ward. Then the 1st goal for the robot is to enter the ward. Once it enters it moves ahead to its 2nd goal which is the 1st table. Once it has reached to the table it opens its arm with the medicine on it and places it over the table by initiating the carry function.

```
def carry(th1,th2,medicine):
    for x in range(300):
        grip = rospy.ServiceProxy('/gazebo/set_model_state', SetModelState)
        arm = rospy.ServiceProxy('/gazebo/get_link_state', GetLinkState)
        state = arm("hospital_robot::link_3","")
        state_msg = ModelState()
        state_msg.model_name = medicine
        state_msg.pose.position.x = state.link_state.pose.position.x
        state_msg.pose.position.y = state.link_state.pose.position.y
        state_msg.pose.position.z = state.link_state.pose.position.z
        grip_state = grip(state_msg)
        pub_1.publish(th1)
        pub_2.publish(th2)
        rospy.sleep(0.01)
```

**Figure 30: carry function.**

After finishing 2nd goal (doing 1st table) the robot moves into the middle of the ward which is its 3rd goal and then moves towards the 4th goal which is the 2nd table. After reaching at its 4th goal the robot does the same operation which is collecting medicine from its drawer and puts it over the table. After finishing the task on the 2nd table, the robot moves towards its 5th goal which is exiting the ward. The Bumblebot does operate the tasks similarly for all the wards and once it reaches its final goal which is end of the hospital wards it returns to its starting position. All the parameters for each goal were collected form robot navigation. The whole main function is displayed below:

```python
def main():
    arm(1.57,3.14) # close arm

    goal(1.8,2.7,1.57) #Enter into ward_1
    rospy.sleep(0.1)
    goal(-0.8, 5.48, 0) #table_1
    arm(0.2,2.60)       #open arm
    carry(1.57,0,"medicine_1")   #above table
    arm(1.57,3.14)   #close
    goal(2.6, 7, 0.8) #middle of room
    goal(5.061683, 6.663107, 2.653774)    #table_2
    #goal(2.6, 6.01, 3.14)
    arm(0.2,2.60)   #open arm
    carry(1.57,0,"medicine_2")   #above table
    arm(1.57,3.14)   #close
    goal(2,0,0) #ward 1 exit
    rospy.sleep(0.1)


    goal(9.6, 2.66, 1.57) #Enter into ward_2
    rospy.sleep(0.1)
    goal(6.8, 4.08, 0) #table_3
    arm(0.2,2.60)       #open arm
    carry(1.57,0,"medicine_3")   #above table
    arm(1.57,3.14)   #close
    goal(10, 5.6, 0.8) #middle of room
    goal(12.68, 6.1, 3.14)    #table_4
    arm(0.2,2.60)   #open arm
    carry(1.57,0,"medicine_4")   #above table
    arm(1.57,3.14)   #close
    goal(10.1,0,0) #ward 2 exit
    rospy.sleep(0.1)


    goal(17.2, 2.66, 1.57) #Enter into ward_3
    rospy.sleep(0.1)
    goal(14.5, 4.24, 0) #table_5
    #goal(14.5, 4.26, 0)
    arm(0.2,2.60)       #open arm
    carry(1.57,0,"medicine_5")   #above table
    arm(1.57,3.14)   #close
    goal(17.7, 5.6, 0.8) #middle of room
    goal(20.34, 6.11, 3.14)    #table_6
    arm(0.2,2.60)   #open arm
    carry(1.57,0,"medicine_6")   #above table
    arm(1.57,3.14)   #close
    goal(17.26,0,0) #ward 3 exit
    rospy.sleep(0.1)

    goal(22,0,1.57) #end of hospital
    goal(11,-0.5,3.14)
    goal(0,0,0)      #return to orignal position

    rospy.signal_shutdown("Done")

if __name__ == '__main__':
    try:
        while not rospy.is_shutdown():
            main()
            rate.sleep()
    except rospy.ROSInterruptException:
        pass
```

**Figure 31: main function**

## 5.6 Project Limitations:

The project might have these possible limitations in these following sectors.

**Ethical Limitations:**

- Patient data and information privacy issues.
- There are safety issues with using robots in hospitals.
- Potential ethical issues with robots taking the place of humans as employees.

**Environmental Limitations:**

- How the actual robot component manufacturing and disposal affect the environment.
- The robot's energy usage and carbon footprint.

**Financial Limitations:**

- Limited budget to start work for the complete project.
- High cost for powerful computer to run the system comfortably and smoothly.

**Time Limitations:**

- Limited time for designing, developing and testing the robot.
- Delayed timelines due to unexpected issues or challenges.

**Policies Limitations:**

- Regarding the usage of robots in healthcare settings, adherence to hospital regulations and procedures is required.
- Regulations governing the use of medical equipment in a hospital setting.

**Human resources Limitations:**

- Limited access to trained labour for programming and developing robot.
- Staff acquisition and retention for the project.

# 6. Results:

The project's results are encouraging and show the feasibility of this project. The robot was able to successfully navigate and accomplish a series of tasks, including picking up and delivering medication to several hospital wards, thanks to the creation of a simulated environment in Gazebo. Using data from its lidar sensor, the robot created an operation map and was able to correctly identify and find its target delivery location and administer the medication to the intended patient. The robot was able to choose the optimal course of action. In this result section all the project accomplishments are discussed. Completion of the different parts of the project was discussed first. Then the robot operation was illustrated.

## 6.1 Completed Virtual Environment:

Virtual hospital environment was a crucial part of the robot. To achieve this, a number of elements are added to the virtual hospital environment, such as doors, floors, and walls, which are created using the proper models and textures. The hospital plan, including patient rooms, hallways, was designed and implemented in the virtual environment created in SweetHome3D. A launch file was later created to launch the virtual environment with the robot into Gazebo. To guarantee that the robot can navigate and carry out activities effectively, the virtual hospital environment needed to be an exact replica of the actual hospital setting. And it was designed keeping that idea under consideration. To ensure that the robot cold travel and carry out activities effectively, it had to be a true representation of the surroundings. For it to be accurate and dependable in all settings, continuous testing and validation were done.

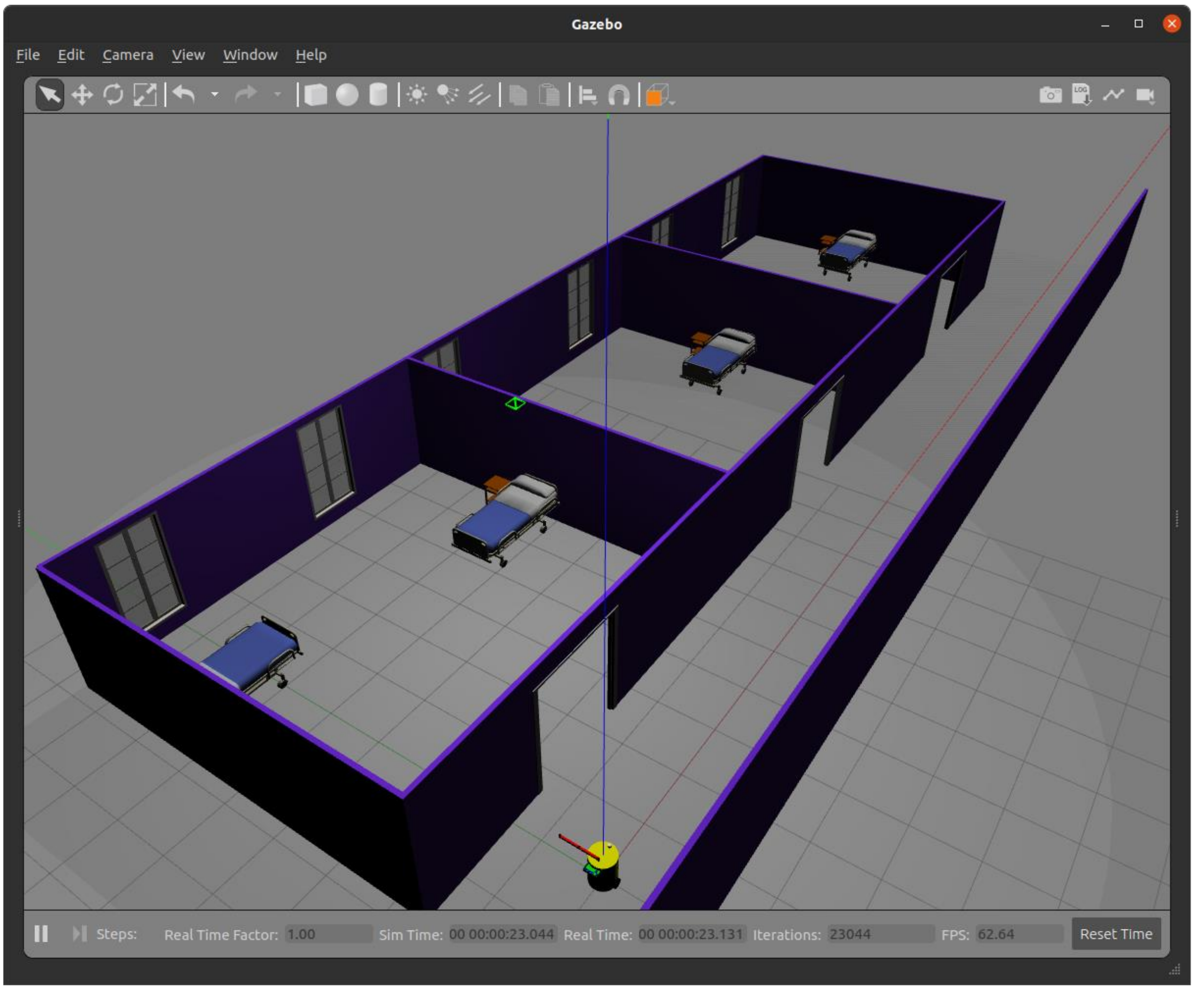


**Figure 32: Completed Virtual World.**

## 6.2 Accomplished Robot:

The robot for this project went through various rounds of development and testing before being finished. First, the robot's mechanical design was developed while taking the needs of the hospital environment into account. The design was then created and put together. The URDF file of the robot was made correctly and later the robot was visualize on Rviz. A launch file was developed for launching the robot into Gazebo. Python coding was also used to create the software used to drive the robot, which also included Rviz, Gazebo, and ROS. A variety of tests were run on the robot once it had been fully assembled and tested to gauge how well it would perform in a hospital simulation. During these trials, the robot's ability to find its way around, pick up and deliver medication, and avoid obstacles were all put to the test. Modifications to the software were made in order to fix any problems that surfaced during testing. The completion of Bumblebot was a significant achievement at the end.

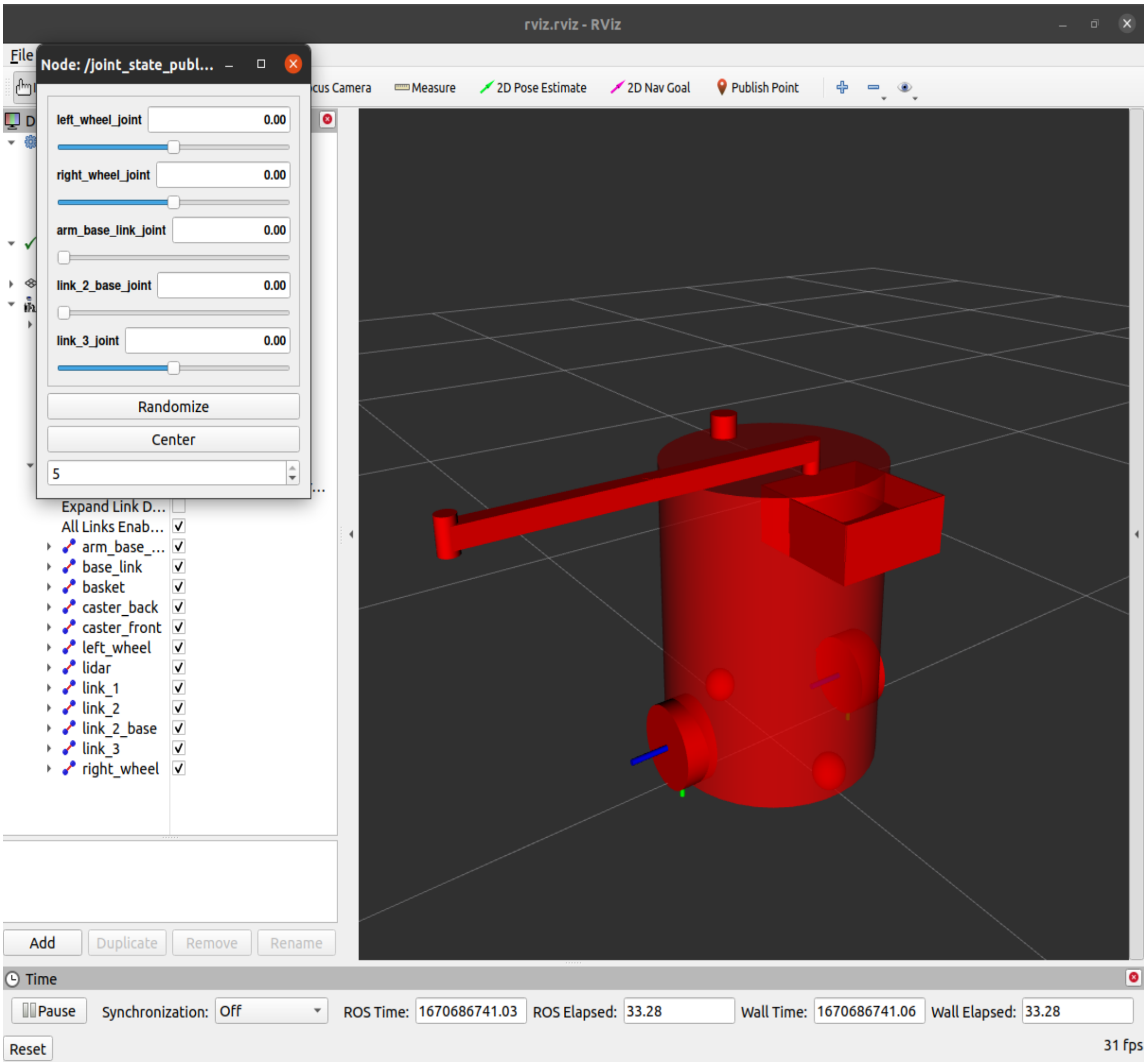


**Figure 33: Complete Bumblebot.**

## 6.3 Mapping Completion:

A full map for this project was made using a mapping and localization algorithm, such as the simultaneous localization and mapping (SLAM) algorithm. The algorithm used data from the robot's lidar to build a map of the environment and simultaneously determine the robot's position within that environment. In the case of this project, the robot was equipped with lidar to capture data about the hospital environment. This data was then fed into the SLAM algorithm, which processed the data and built a map of the environment in real-time. The SLAM algorithm then used this map to determine the robot's position as it navigated through the hospital environment. Once the map was created, it was used by the robot for navigation, as well as for planning and executing tasks within the environment. The map was also updated in real-time as the robot continued to explore the environment and gather new sensor data, ensuring that the robot always had an accurate representation of the environment in which it was operating.

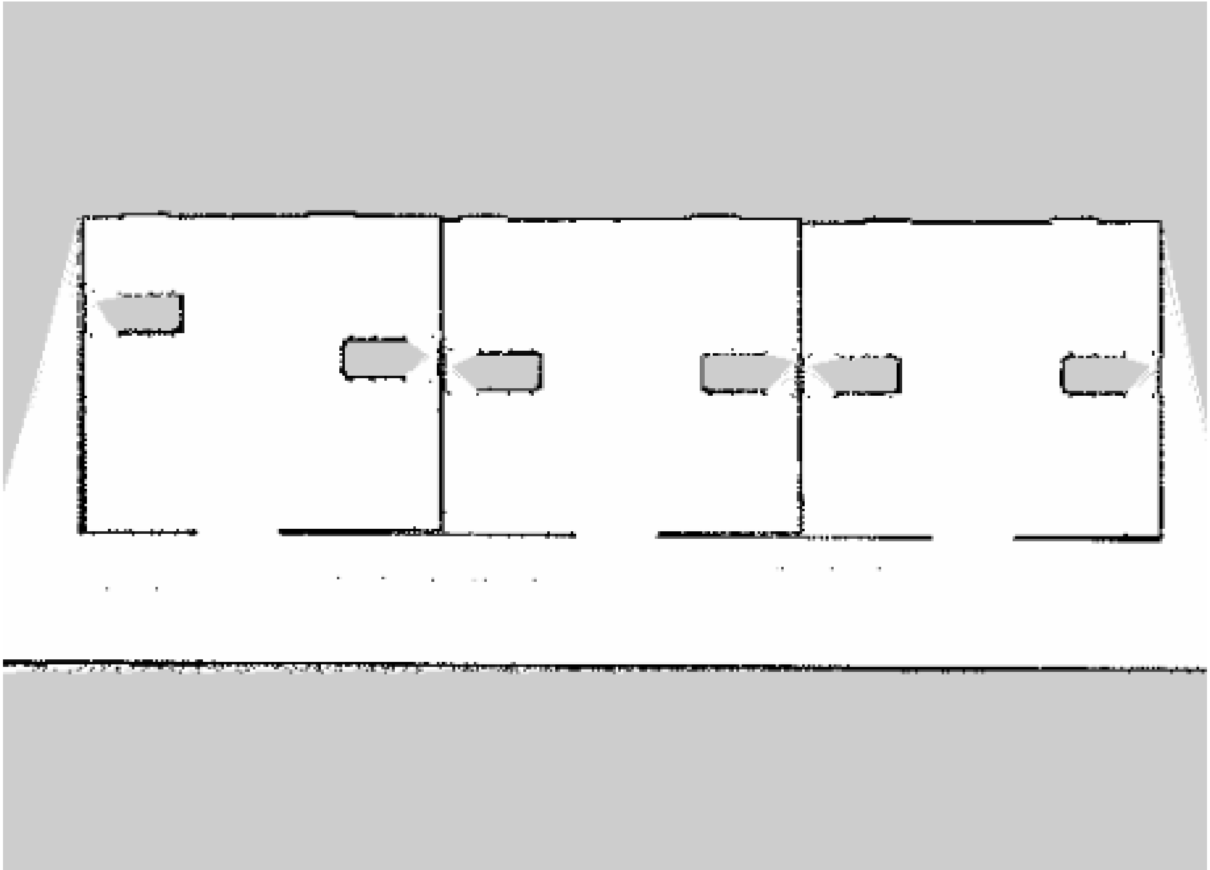

**Figure 34: Complete map for navigation.**

## 6.4 Bumblebot's complete operation:

Below, the complete delivery operation is illustrated step by step:

**Step-1:** Bumblebot is starting from its starting point.

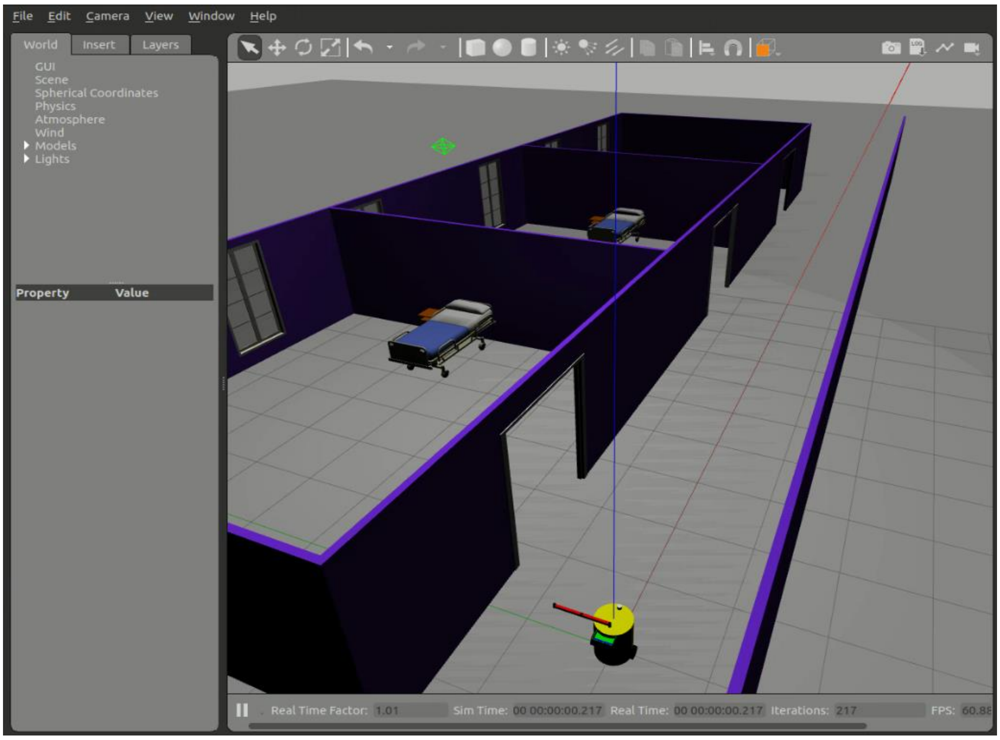


**Figure 35: Step-1.**

**Step-2:** The robot is entering into the 1st ward.

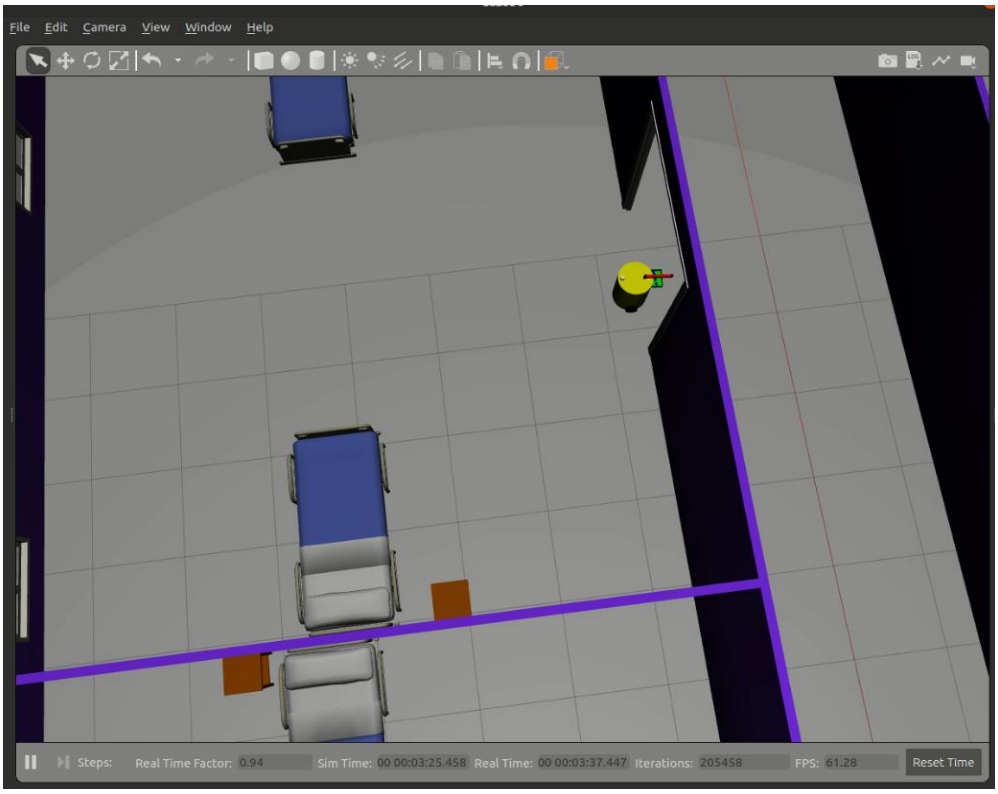


**Figure 36: Step-2.**

**Step 3:** The robot is approaching to the table near the 1st bed.

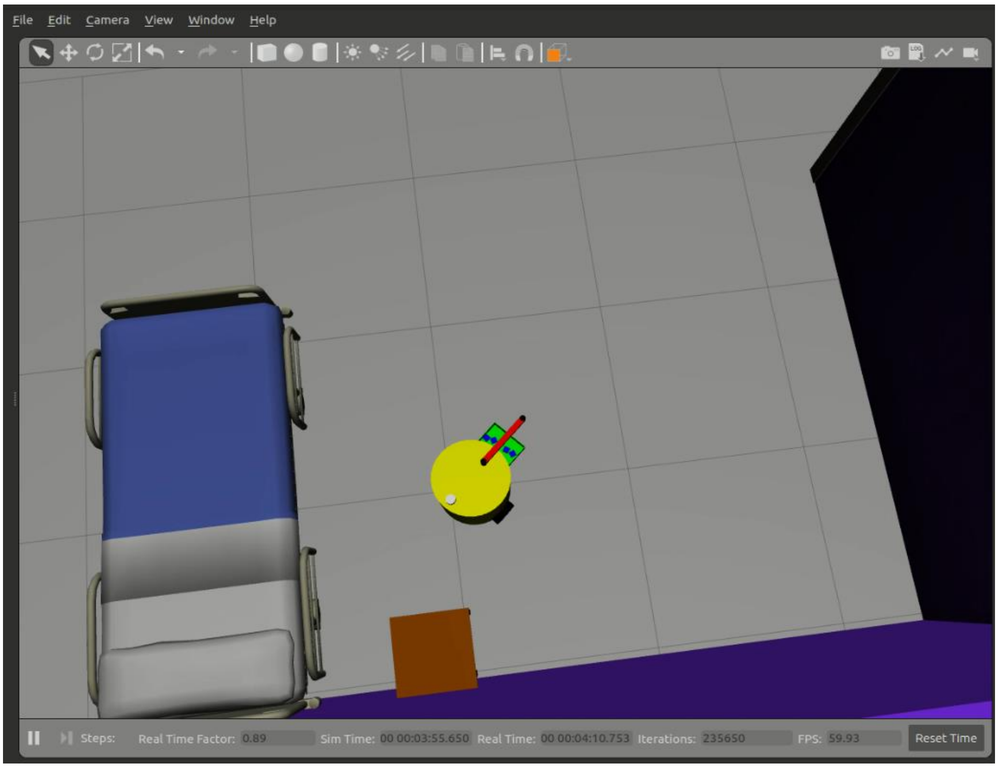


**Figure 37: Step 3.**

**Step 4:** The robot is grabbing the medicine from its basket.

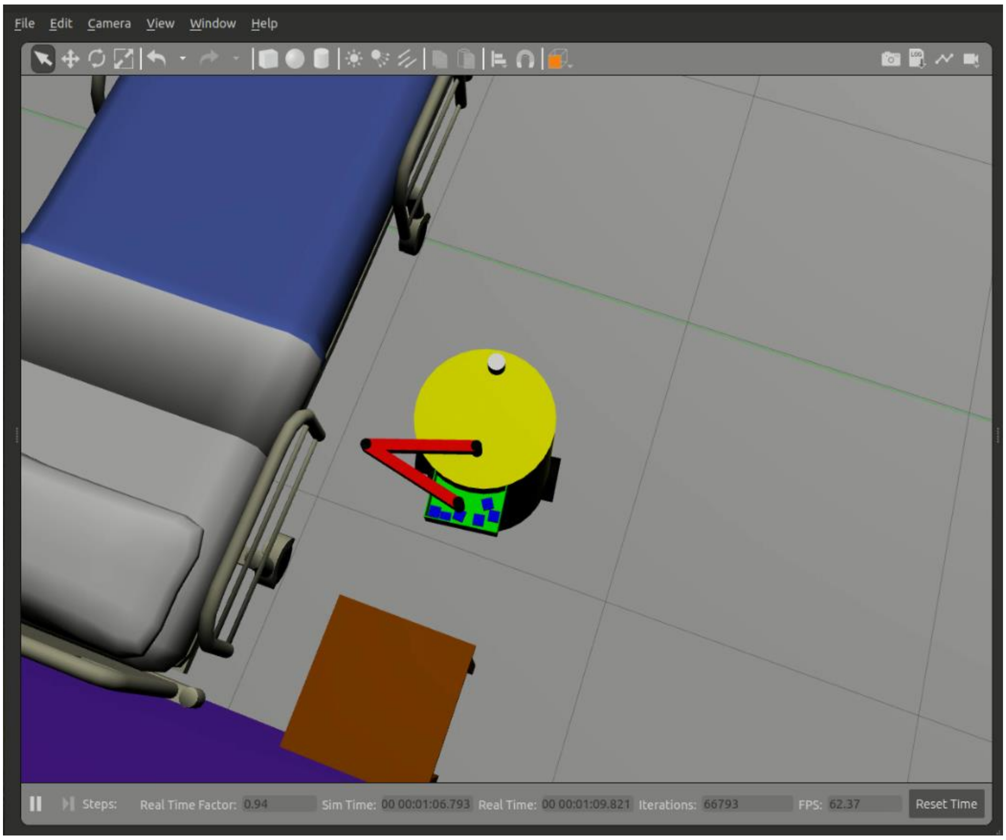


**Figure 38: Step 4.**

**Step-5:** Bumblebot is placing medicine over the table.

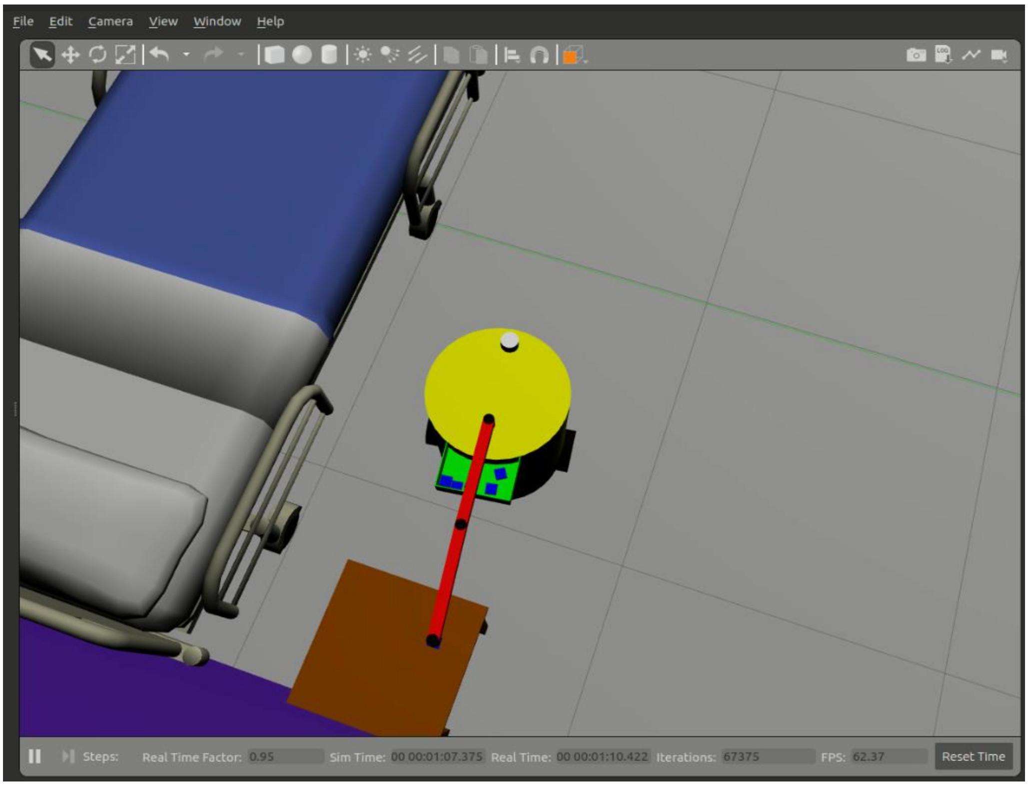


**Figure 39: Step-5.**

**Step-6:** Medicine is placed over the table.

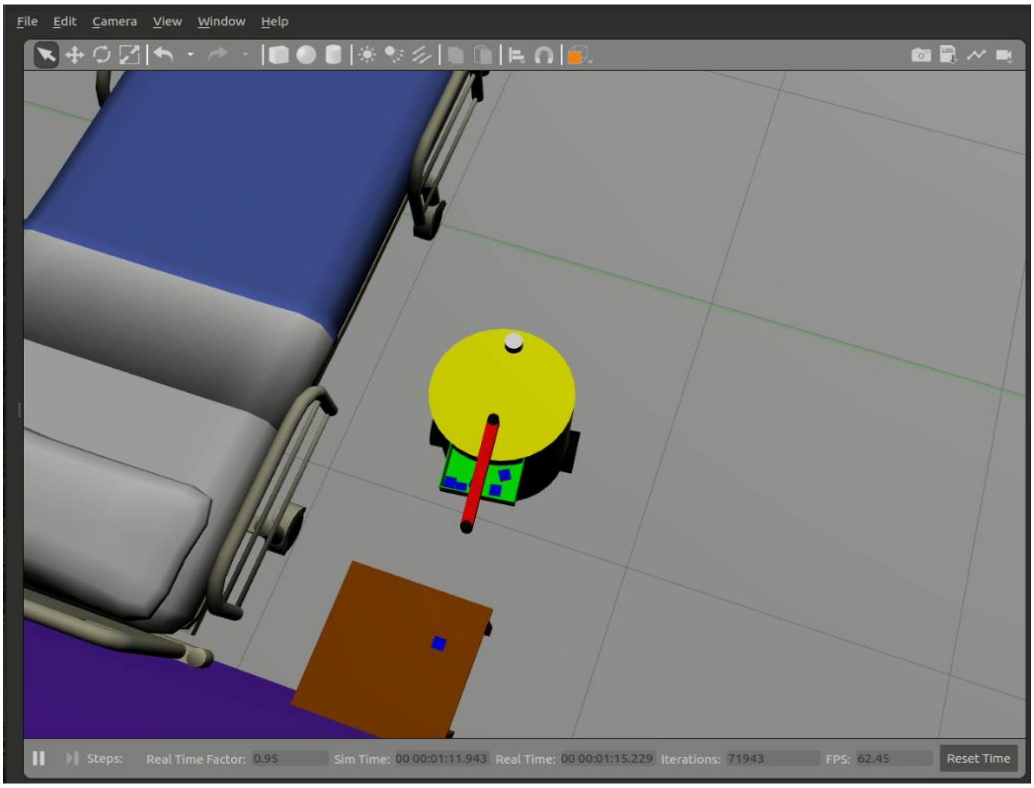


**Figure 40: Step-6.**

**Step-7:** Bumblebot is moving towards next bed.

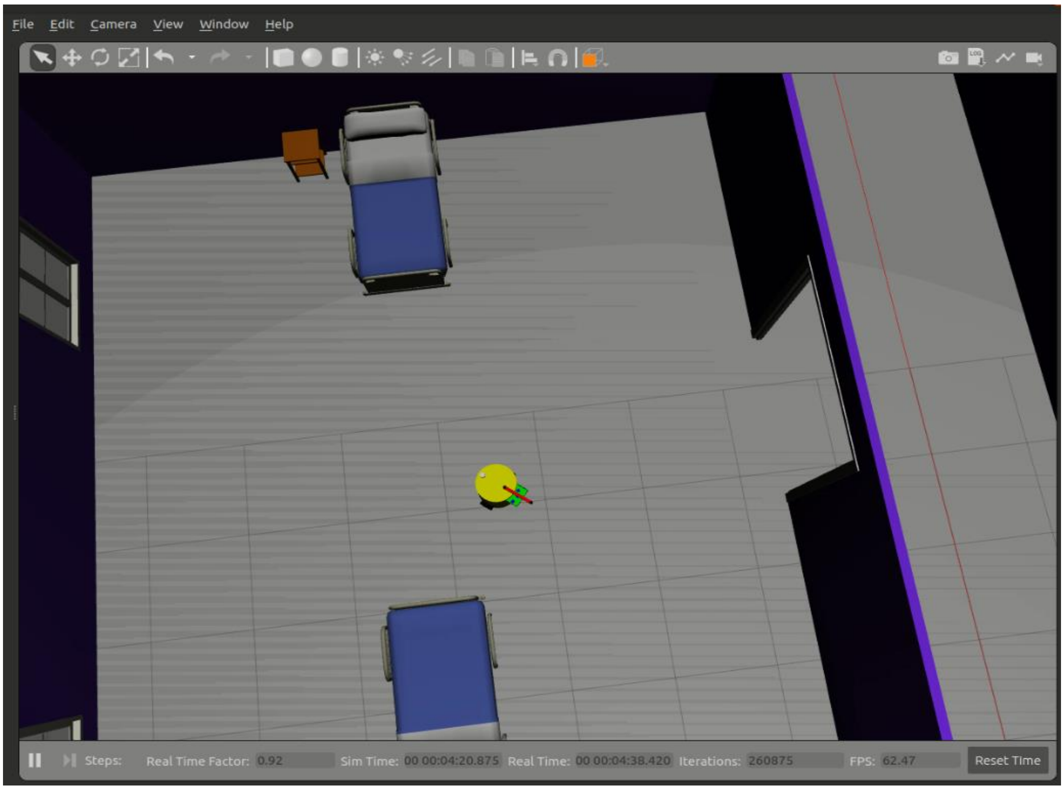


**Figure 41: Step-7.**

**Step-8:** Robot is moving towards next ward.

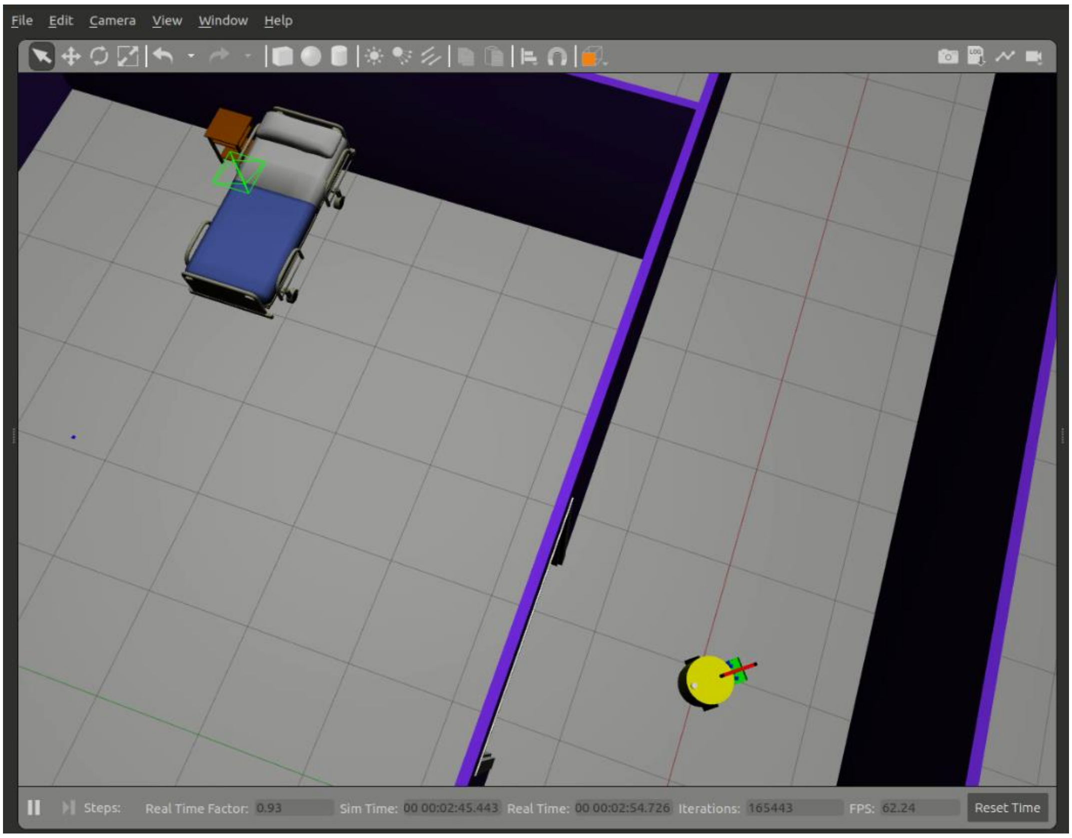


**Figure 42: Step-8.**

**Step-9:** Robot is returning to its starting position.

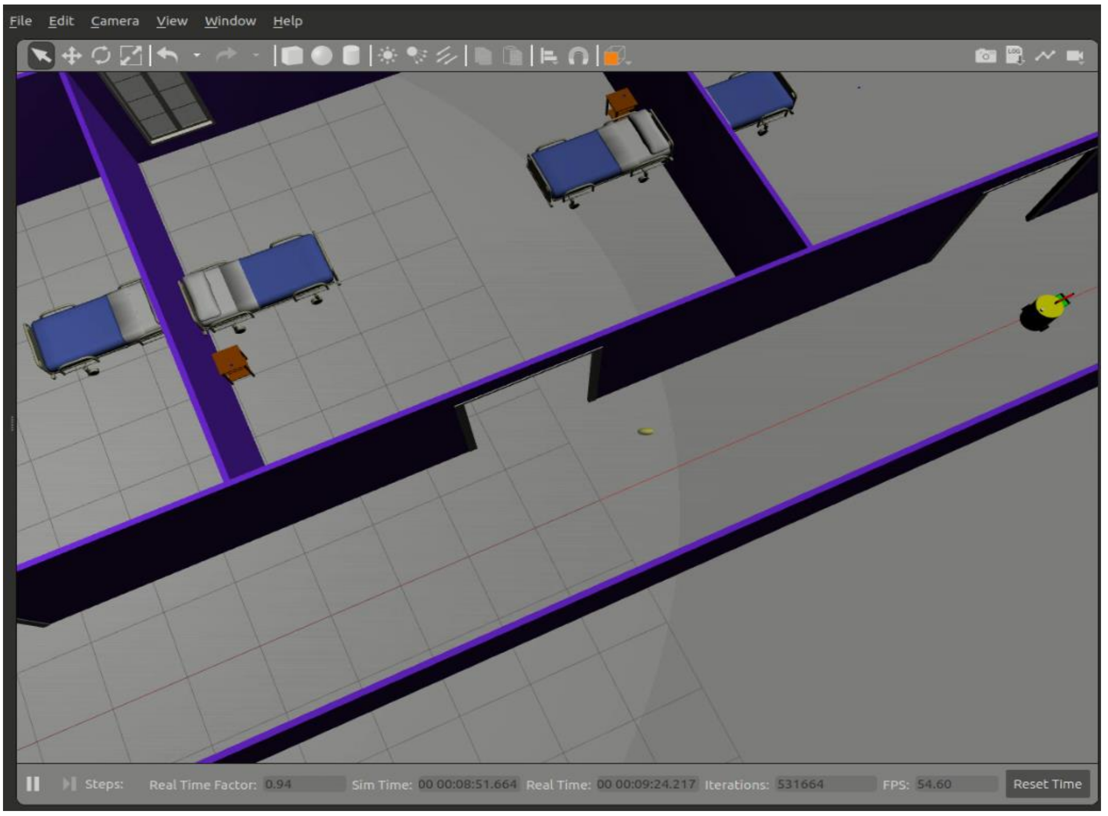


**Figure 43: Step-9**

# 7. Analysis and Scientific Argument:

## 7.1 Development and Coherence of Arguments from the literature:

In the literature review, various research related with this project were covered. Each study offered a unique method and method for modelling various tasks, including exploration, mapping, manipulation, transportation, and coordination. Each study outlined its justifications and supporting data to back up its methodology. For instance, Niko et al. (2019) demonstrated the efficiency of their method for replicating actual exploration circumstances in Gazebo by performing simultaneous localization and mapping (SLAM) using the ROS navigation stack and the GMapping algorithm. Like this, Gao et al. (2020) demonstrated the efficiency of their technique by simulating real-world manipulation scenarios in Gazebo using the MoveIt! package to plan and carry out manipulation activities. Overall, the literature review offered a cogent and well-structured argument, with each study adding to the overall image of employing simulation robots in Gazebo for diverse tasks. The research clearly communicated their conclusions and offered proof to back up their assertions. Additionally, the literature review identified the typical tools and techniques employed by researchers for simulating various tasks, emphasising the effectiveness of these tools and techniques in Gazebo.

## 7.2 Comparative analysis between project findings and literature review:

In this project, determining the degree of alignment and spotting any differences, the results of this project can be compared to the existing literature review. The comparative analysis can assist in identifying the gaps in the current body of knowledge and possible areas for further study.

For instance, various research has emphasised the value of creating a precise navigation and localization system for Gazebo simulation robots in the literature review. The success of the navigation and localization technology employed by the BumbleBot was evaluated by comparing the project results with those from this research. If the results showed a high degree of accuracy and reliability, they could reinforce the body of literature already in existence. On the other side, if there were any substantial differences, it may draw attention to areas that needed improvement or additional research for Bumblebot.

Similar comparisons can be made between the project's findings and the body of current literature regarding the robot's aptitude for a variety of tasks. The literature that highlights the potential of simulation robots in performing a variety of tasks in a simulated environment, for example, can be supported if the project's findings demonstrate that the simulation robot can perform tasks like grasping, pushing, and lifting objects effectively.

In summary, a comparison of the project results with the findings from the literature review revealed the strengths and weaknesses of the existing literature, pointed out potential directions for further study, and shed light on the efficiency of the simulation robot created for this project.

## 7.3 Development and quality of the scientific argument:

The development and quality of the scientific argument in this project depend on several factors, including the research question, methodology, data analysis, and interpretation of the findings.

In this project, the creation of a ROS-operated medicine delivery robot in a hospital environment is the main research topic in this project. To address this question successfully, an interdisciplinary strategy integrating expertise in robotics, software engineering, and healthcare is necessary. The approach taken for this project combines simulation with ROS programming, testing, and validation, all of which are done in a simulated environment. The data analysis includes gathering and analysing information about the robot's performance,

such as its accuracy in localization and navigation, versatility in task execution, and effectiveness in drug delivery. Based on a comparison between the results and the project's goals and the body of previous research, the results are then interpreted.
Overall, the project's scientific justification was of a very high calibre because it employed an exacting and methodical procedure to respond to the study issue. To ensure the Bumblebot's accuracy and dependability in a medical environment, the project used cutting-edge technologies and methodologies to create and test it. The conclusions, which are backed by the body of existing research, were based on a careful examination of the data gathered. The results were objectively interpreted by the project in a logical and scientifically sound manner, emphasising the simulation robot's potential to enhance healthcare services. The scientific argument in this project was well developed and of high quality, offering important insights into the creation of this project. The research highlights the value of an interdisciplinary approach as well as the application of contemporary robotics and software engineering tools and techniques.

## 7.4 Evidence of the ability to evaluate information and synthesise conclusions:

Any research project must have the capacity to evaluate data and combine findings. There were numerous instances of this ability being displayed in this project. For instance, during the literature review, data from multiple sources was assessed and synthesised to create a thorough grasp of the research that had already been done on simulation robots and virtual worlds. Additionally, information from various sources was gathered, assessed, and synthesised during the data analysis phase in order to draw conclusions about the success of the robot's navigation and task completion abilities.

Additionally, the robot's performance was optimised during development and testing by comparing various algorithms and techniques. It was up to the author to assess the information that was available on various techniques and decided which ones were most suited to the project's goals. A conclusion was reached regarding the most efficient methods and how they affected the performance of the robot through testing and evaluation.

Last but not least, the project's conclusions and recommendations show how to analyse data and make generalisations. Based on the project's findings, suggestions were made for further study, creation, and use of simulation robots in virtual settings. Based on the evaluation and synthesis of the project's findings along with the body of prior research in the area, several suggestions were made.

Through its phases of literature review, data analysis, algorithm development, and recommendation, the project as a whole demonstrates a strong capacity to evaluate information and synthesise conclusions.

## 7.5 Innovation style: problem solving, or step-by-step learning through established techniques:

Identification and resolution of issues or challenges through a creative and analytical process are key components of the innovative style of problem-solving. Instead of depending on tried-and-true approaches or processes, this strategy emphasises the discovery of novel solutions to issues. The technique to solving problems requires a flexible and adaptive mentality that enables experimentation and iteration to develop and refine solutions until a suitable result is obtained.

Throughout the whole project development stage different problem-solving methods were used. Making this whole project successfully complete was difficult. While tackling the problems there was also necessity of self-involvement in establishing techniques for learning step by step for each allocated work. This requires an innovative approach to problem-solving, as the project involves multiple challenges that require creative solutions. The project requires identifying and addressing specific problems such as navigation, task planning and self-relocating.

During the learning process many skills were developed step by step. Such as learning how to create a model robot from creating a URDF file to visualize it. Another possible example was, while creating the map many test runs were executed and finally after a complete run and getting all the data from the sensor a fully built map was made.

# 8. Evaluation and Accomplishment:

## 8.1 Critical appraisal and evaluation of the project and process:

Critical appraisal and evaluation of a project and its process is a crucial step in making sure a project achieves its aims and objectives requires thorough examination and critical appraisal of the project's methodology. It involves a detailed evaluation of the project's performance, considering all of its advantages, disadvantages, opportunities, and threats.

Assessing this project's success in achieving its objectives and goals was an important part of its evaluation. To did this, the project's results were examined and contrasted with its initial goals. For instance, the main goal of this project was to build a fully functional robot that could move around a hospital environment and gave medications to patients in a secure and effective way. Because of this, the critical evaluation was concentrated on assessing the robot's performance in terms of navigation, delivery effectiveness, and safety.

Another important aspect of critical appraisal and evaluation was to analyse the project's process, including its planning, design, development, and testing phases. This involved assessing the project's timeline, budget, and resource allocation, as well as the effectiveness of the project management approach. For example, in this project, the critical appraisal was evaluating the efficiency of the project planning, the effectiveness of the software development process, and the quality of the testing and validation procedures.

Critical appraisal and evaluation were essential steps in ensuring the success of this project. It helped to identify areas of improvement and opportunities for future development. Therefore, it was crucial to conduct a thorough and comprehensive evaluation of the project to ensure that it met its goals and objectives effectively.

## 8.2 Reflection of self-development whilst conducting the project:

The process of carrying out this project provided a precious chance for personal improvement in a variety of ways. Here are some potential self-development areas that was considered:

- **Technical skills development:** Many different technical skills, including programming, electronics, and mechanical design, were implemented to create Bumblebot which could function in a virtual hospital setting. By working with different technical tools like ROS, Gazebo simulator, gmapping package of Ros the project maker gained expertise. Working on programming languages like Python was a great opportunity to get deeper understanding in this programming language.
- **Teamwork with project supervisor:** Throughout the whole period of developing this project, regular contact with project’s supervisor was made. Any difficulties found at any part of development were immediately reported to the supervisor and prompt actions were taken to solve the problem.

- **Development on project management skills:** This project was needed to manage effectively, which meant planning, organising, and tracking progress were a necessary part and were effectively taken care of. Abilities and expertise in managing time, resources, and priorities along the course of the project were also acquired.
- **Self-awareness and personal growth:** Conducting this project provided an opportunity for self-discovery and personal development. Personal qualities, limitations, and potential growth areas were discovered through reflection on this project.

## 8.3 Achievements and shortcomings of the project:

Every goal listed in the introduction section has been accomplished throughout the duration of this project. The accomplishment of these objectives has been secured by the meticulous and methodical approach used in each phase of the project, from design and development to testing and evaluation. Below table illustrates all the completed objectives of this project.

| Objective | Status |
|---|---|
| Creating a dependable, effective autonomous robot that can transport supplies and medicines in a clinical or hospital setting. | Completed |
| Designing a user-friendly interface so that medical staff may easily configure, monitor, and control it. | Completed |
| Make sure the software is reliable and able to survive frequent use in healthcare environments. | Completed |
| Adding cutting-edge sensors and algorithms to improve navigation, obstacle avoidance, and responsiveness to changing situations. | Completed |
| Promoting the acceptance and further development of autonomous robots in healthcare by disseminating research findings and suggestions. | Completed |
| The Bumblebot robot is designed and built within seven months starting in October 2022, with regular progress updates and milestones set throughout the project timeline. | Completed |
| The whole project was developed inside the project budget and met all the criteria according to the UK SPEC. | Completed |

To summarise the overall build and features completed during the project, the following main points are listed:
**Designing and building a simulated hospital environment:** A simulated hospital environment with three wards was designed and built successfully.
**Designing and building a simulated robot with a personalized URDF file:** A URDF file was developed with all planned links and joints successfully and later visualize through Rviz.
**Development of a map from SLAM gmapping system for robot's localization:** After the robot completed a full run over all the wards a map was generated successfully for further autonomous operation. All the information was collected from the robot's lidar sensor.
**Robot task planning:** A python script was developed for robot's medicine delivery task and later successfully implemented.
**Autonomous robot operation:** The Bumblebot did an auspicious run through all of the wards autonomously while delivering the medicines and at the end returned to its initial position without crashing.

## 8.4 Further development and recommendations:

Here are some suggestions for the ROS-operated medicine delivery robot project in terms of further study, development, and research:

- **Strengthening the robot's capabilities:** The robot's present model is made to move around a medical setting and give patients their medications. Future research might concentrate on developing the robot's capacity to perform additional duties, such delivering food or medical supplies.
- **Testing in real-world hospital environments:** Although the current project has undergone simulation testing, future study should concentrate on testing the robot in actual clinical settings to assess its utility and viability.
- **Integration with hospital information systems:** The robot could retrieve and distribute medications to patients on its own, lessening the effort of hospital employees and assuring the precision and safety of medication administration. This would be possible by integrating the robot with electronic health records and pharmacy management systems.
- **Human-robot interaction:** It is crucial to think about how these technologies may affect the experiences of patients and employees as the usage of robots in healthcare settings increases. Future studies could investigate ways to enhance the robot's interaction with patients and medical personnel as well as the ethical and societal implications of the technology**.**
- **Sensor Development:** In future a camera can be implemented along with the lidar for using advanced object detection methods.
- **Optimizing navigation and obstacle avoidance**: While the current navigation system works, future research may investigate more sophisticated navigation and obstacle avoidance methods. More convenient object detection methods like YOLO can be implemented and machine learning algorithms or more advanced mapping and localization techniques can be used in this.
- **Cost-effectiveness:** The fact must be kept on mind when making this robot for real-world application is that the robot is cost-effective for the buyers. Making the robot with renewable materials can help keeping its price low and make it more practical as well inexpensive for hospitals to use.

## 8.5 Relating the project to UK-SPEC competencies:

This project relates to several competencies outlined in the UK-SPEC (UK Standard for Professional Engineering Competence) framework.

Firstly, the project necessitates the use of engineering principles to address challenging issues in mechanical design, robotics, and control systems. This exhibits the ability to utilise engineering expertise and insight to create workable solutions.

Secondly, to design and test the performance of the robot, the project makes use of simulation tools like Gazebo and Rviz. This demonstrates the capability of developing, simulating, and testing engineering solutions using the proper approaches and tools.

Thirdly, the project requires the creation of a navigation system that enables the robot to travel independently and carry out duties, demonstrating the ability to plan and create complicated systems.

Finally, the initiative has broader societal and commercial consequences in the areas of ethics, the environment, finances, and human resources. The project abided by moral principles, including protecting patient privacy and making sure that patients are safe.

Environmental consequences, including energy use and trash reduction was considered during the robot's design and operation. Along with the necessary human resources to design and successfully carry out the project, the project's cost and financial ramifications were also considered.

In short, during the development process for this project several competencies outlined in the UK-SPEC framework, including applying engineering knowledge, using appropriate techniques and tools, designing complex systems, and considering wider social and industrial implications.

# 9. References:

Organisation: The Health Foundation.
Title of Website: NHS staff shortages put long-term vision for primary and community care at risk.
Available from: https://www.health.org.uk/news-and-comment/news/nhs-staff-shortages-put-long-term-vision-for-primary-and-community-care-at-risk
[Accessed 23/10/2022]

Organisation: Wikipedia.
Title of Website: Workplace robotics safety.
Available from: https://en.wikipedia.org/wiki/Workplace_robotics_safety
[Accessed 23/10/2022]

Organisation: AVG in Hospital.
Title of Website: Autonomous Mobile Robots Disrupting Healthcare Automation.
Available from: https://www.agvnetwork.com/Automation-Hospitals-AGV-Autonomous-Mobile-Robots
[Accessed 01/11/2022]

Organisation: The Wired.
Title of Website: This Incredible Hospital Robot Is Saving Lives.
Available from: https://www.wired.com/2015/02/incredible-hospital-robot-saving-lives-also-hate/
[Accessed 02/11/2022]

Organisation: Panasonic Group.
Title of Website: Panasonic Autonomous Delivery Robots - HOSPI - Aid Hospital Operations at Changi General Hospital
Available from: https://news.panasonic.com/global/topics/4923
[Accessed 04/11/2022]

Author: Liu Z., Shi Y. & Chen Y. (2018).
Title of Research: Design and simulation of a hexapod robot based on SolidWorks and ROS.
Available from: In 2018 IEEE International Conference on Information and Automation (ICIA) (pp. 1368-1372). IEEE.
[Accessed 04/11/2022]

Author: Fankhauser P., Bloesch M., Hutter, M., & Siegwart R. (2016).
Title of Research: ROS-based SLAM for a Gazebo-simulated mobile robot in 2D and 3D environments: An experimental study
Available from: In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 31-38). IEEE.
[Accessed 04/11/2022]

Author: Song S., Lee S., Lee H., & Ryu J. (2020).
Title of Journal: Design of a drone's navigation system in a dynamic environment using Gazebo.
Available from: Journal of Intelligent & Robotic Systems, 98(2), 431-444.
[Accessed 04/11/2022]

Author: Niko M., Harju M., & Kuusniemi H. (2019).
Title of Journal: Simulating robot exploration and mapping in Gazebo.
Available from: In Proceedings of the 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA) (pp. 319-324). IEEE.

[Accessed 04/11/2022]

Author: Gao, W., Lu, B., & Liu, H. (2020).
Title of Journal: Simulation of robot manipulation tasks in Gazebo.
Available from: In Proceedings of the 2020 International Conference on Robotics and Automation Engineering (ICRAE) (pp. 161-166). IEEE.
[Accessed 04/11/2022]

Author: Li, S., Chen, X., Li, H., & Li, Y. (2021).
Title of Journal: Simulation of mobile robot transportation task based on Gazebo.
Available from: Journal of Physics: Conference Series, 1835(1), 012031.
[Accessed 04/11/2022]

Author: Murty P. K., Pati U. C., & Roy P. P. (2018).
Title of Journal: Dynamic mapping of indoor environment using gmapping algorithm.
Available from: Journal of Intelligent & Robotic Systems, 92(1), 1-17.
[Accessed 04/11/2022]

Author: Huang K., Liu F., Xie D., & Luo, H. (2019).
Title of Journal: A FastSLAM Algorithm for Large-Scale Environments Based on Memory Optimization.
Available from: Journal of Intelligent & Robotic Systems, 93(3), 409-425.
[Accessed 04/11/2022]

Organisation: Aeton
Title of Website: Change Healthcare
Available from: https://www.aethon.com/tug-autonomous-mobile-robot/
[Accessed 23/10/2022]

Organisation: Savioke
Title of Website: Relay Robots.
Available from: https://www.savioke.com/relay-robot/
[Accessed 23/10/2022]

Organisation: ROS.org
Title of Website: Building a Visual Robot Model with URDF from Scratch
Available
from: http://wiki.ros.org/urdf/Tutorials/Building%20a%20Visual%20Robot%20Model%20with%20URDF%20from%20Scratch
[Accessed 23/10/2022]

Organisation: ROS.org
Title of Website: Using slam navigation
Available from: http://wiki.ros.org/cob_tutorials/Tutorials/Navigation%20%28slam%29
[Accessed 23/10/2022]

Organisation: ROS.org
Title of Website: navigation
Available from: http://wiki.ros.org/navigation
[Accessed 23/10/2022]

Organisation: ROS.org
Title of Website: gmapping
Available from: http://wiki.ros.org/gmapping
[Accessed 23/10/2022]

Organisation: ROS.org
Title of Website: Program your robot with the Python API
Available from: http://wiki.ros.org/pilz_robots/Tutorials/ProgramRobotWithPythonAPI
[Accessed 23/10/2022]