

##### **NIT6150**

##### **Advanced Project**

##### System Design and Approach

##### **2D SLAM-based Childcare Robot**

**Team Leader:** Mary Yazbek (s4674864)

**Team Members:** Meghna Meghna (s4683216)

John Machado (s4685542)

**Client:** Dr. Wenjie Ye  
VU Sydney

**Supervisor:** Dr. Wenjie Ye

VU Sydney

**Coordinator:** Dr. Wenjie Ye

College of Engineering and Science

Victoria University, Melbourne

Table of Contents

[Introduction 2](#_Toc174878672)

[System Development Approach 2](#_Toc174878673)

[Functional Requirements 2](#_Toc174878674)

[Use Case 3](#_Toc174878675)

[Non-Functional Requirements 3](#_Toc174878676)

[System Navigation and Interface 3](#_Toc174878677)

[System navigation 3](#_Toc174878678)

[Interface 4](#_Toc174878679)

[Cost Estimate 6](#_Toc174878680)

[Schedule 8](#_Toc174878681)

[Considerations 13](#_Toc174878682)

[Human Factors and Privacy Preserving 15](#_Toc174878683)

[Risk and Issue mitigation 16](#_Toc174878684)

[Ethical Issues 17](#_Toc174878685)

[References 18](#_Toc174878686)

[Appendix 19](#_Toc174878687)

# Introduction

Falls are a leading cause of injury among children, prompting thousands of emergency visits daily in the U.S. due to accidents involving stairs and windows. This issue highlights the need for enhanced supervision in environments like kindergartens, where constant vigilance is challenging. Leveraging advancements in service mobile robotics, particularly 2D Simultaneous Localization and Mapping (SLAM) technology, presents a viable solution. Our project proposes using 2D SLAM robots equipped with LiDAR sensors for real-time navigation and monitoring within indoor settings such as schools and childcare centers. This approach aims to enhance child safety by identifying and mitigating hazards without compromising privacy, eschewing cameras in favor of non-intrusive sensors.

# System Development Approach

For the development of the SLAM-based mobile robot project, an **Agile development approach** would be most suitable. The Agile approach is well-suited to this project due to its flexibility, focus on user feedback, iterative development, and effective risk management. It will allow the project team to deliver a high-quality product that meets the needs of small childcare centres while staying adaptable to changes and new insights.

**1. Flexibility and Adaptability:**

* **Rapid Iterations:** Agile allows for iterative development, meaning you can develop the robot in incremental stages. This approach enables you to test and refine each component, such as navigation, human detection, and safety features, as they are developed.
* **Responsive to Change:** Given that this project involves integrating advanced technologies, requirements may evolve as the project progresses. Agile's flexibility allows the team to adapt to changes without significant disruptions.

**2. Stakeholder Engagement:**

* **Continuous Feedback:** Agile encourages regular interaction with stakeholders (e.g., childcare centre staff, parents), ensuring that their feedback is incorporated throughout the development process. This engagement helps ensure that the final product meets the actual needs of the users.

**3. Risk Management:**

* **Early Detection of Issues:** The iterative nature of Agile allows for the early identification of potential issues, such as hardware limitations or privacy concerns. Addressing these issues early reduces the risk of major setbacks later in the project.
* **Prioritization of Features:** Agile allows the team to prioritize critical features, ensuring that the most important aspects of the project (e.g., safety features) are developed and tested first.

**4. Quality Control:**

* **Continuous Testing:** Agile involves ongoing testing and validation, which helps maintain high-quality standards throughout the development process. This approach ensures that each component of the robot, from navigation to child detection, functions as intended before the next phase begins.

**5. Efficient Resource Management:**

* **Focused Development:** Agile helps in allocating resources efficiently by focusing on small, manageable tasks. This is particularly beneficial for a project like this, which has several technical components that need specialized attention.

**6. Time Management:**

* **Defined Sprints:** Agile breaks the project into sprints, each with its own timeline and deliverables. This structure helps keep the project on track and ensures timely completion of each phase, aligning with the overall project schedule.

# Functional Requirements

For the project involving the development and deployment of a 2D SLAM-based mobile robot aimed at enhancing safety and operational efficiency in childcare centers, the functional requirements define specific behaviours or functions the system must fulfill. Here’s a detailed breakdown of these requirements:

**1. Navigation and Mobility**

* The robot must autonomously navigate within various indoor environments using its SLAM (Simultaneous Localization and Mapping) capabilities.
* The robot should be able to avoid obstacles and safely maneuver around people, furniture, and other objects without human intervention.
* The robot must return to navigate to and stop at the specified point in space.

**2. Child Monitoring and Safety**

* The robot should be able to guide the child from point A to B without hitting any obstacle, leading the child to safety.

**4. Data Collection and Reporting**

* The robot must collect data on its operational performance for continuous improvement.
* The robot must securely store and transmit data in compliance with applicable privacy regulations.

**5. Maintenance and Updates**

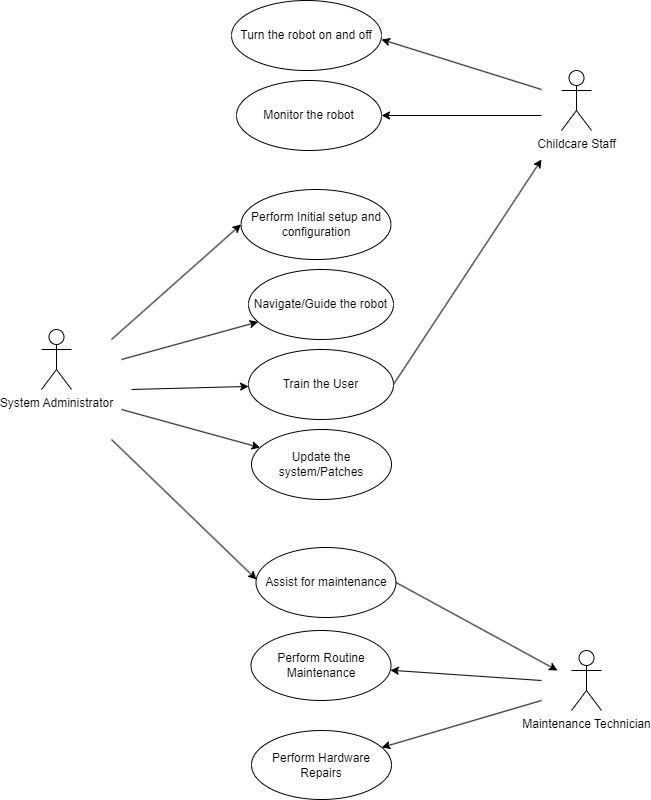
* The robot must be easy to maintain with capabilities for remote diagnostics and software updates.
* The robot must allow for manual override or shutdown by authorized personnel in case of emergencies.

**6. Security and Privacy**

* The robot should ensure that all data collection and monitoring activities comply with child privacy laws and ethical standards.

# Use Case

The overall use case diagram for this system in a small childcare centre will depict the interactions between the three primary actors—**Childcare Staff**, **System Administrator**, and **Maintenance Technician**—and the system. Each actor has specific roles and responsibilities within the system, represented by different use cases.



**Actors and Use Cases**

1. **Childcare Staff**
   * **Turn the Robot On/Off**: This use case involves the basic operation of powering the robot on or off at the beginning or end of the day. It’s a straightforward task but crucial for the robot’s operation.
   * **Monitor the Robot**: Childcare Staff monitor the robot’s activities, such as its navigation through the childcare centre and its interaction with the environment. This includes responding to any alerts generated by the robot.
2. **System Administrator**
   * **Perform Initial Setup and Configuration**: The System Administrator is responsible for the initial setup of the robot, which includes configuring the environment map, setting up user accounts, and configuring system settings to tailor the robot's functions to the specific environment.
   * **Navigate/Guide the Robot**: The System Administrator can manually control or override the robot’s navigation system, directing it to specific locations or overriding its automated paths when necessary.
   * **Train the User**: This use case covers the System Administrator’s responsibility to train Childcare Staff on how to use the robot effectively, including turning it on/off, monitoring, and responding to alerts.
   * **Update the System/Patches**: The System Administrator handles the periodic updates and patches to the robot's software, ensuring it has the latest features, security measures, and bug fixes.
   * **Assist in Maintenance**: The System Administrator may assist the Maintenance Technician during maintenance activities, especially those related to software diagnostics or updates that need to be performed alongside hardware maintenance.
3. **Maintenance Technician**
   * **Routine Maintenance**: This use case involves the regular maintenance of the robot, such as checking the sensors, cleaning the hardware, and performing diagnostics to ensure it is operating correctly.
   * **Hardware Repairs**: If the robot experiences a hardware malfunction or damage, the Maintenance Technician is responsible for diagnosing the issue and performing the necessary repairs or part replacements.

# Non-Functional Requirements

**1. Look and Feel Requirements**

* **User Interface Aesthetics**: The robot's interface, if any (e.g., a display for caregivers), should have a child-friendly and intuitive design, with soft colours and large icons that are easy to understand.
* **Physical Appearance**: The robot should have a non-threatening, appealing design, with a compact size and smooth edges, ensuring it fits seamlessly into childcare environments.

**2. Usability and Humanity Requirements**

* **Ease of Use**: The system should be easy to operate by caregivers with minimal technical knowledge. Clear instructions and simple controls should be provided.
* **Accessibility**: The robot should be accessible to all users, including those with disabilities. This includes easy access to controls, voice commands, and other interactive features.
* **Language Support**: The system should support multiple languages to cater to diverse childcare environments.

**3. Performance Requirements**

* **Response Time**: The system should provide real-time responses, with a maximum delay of 1 second for critical operations like obstacle detection and navigation adjustments.
* **Battery Life**: The robot should have a minimum operational battery life of 8 hours to cover a full day in a childcare setting without needing a recharge.
* **Scalability**: The system should be scalable to operate in various sizes of indoor environments, from single rooms to larger childcare centres.

**4. Operational Requirements**

* **Environmental Conditions**: The robot should operate effectively in typical indoor environments, withstanding temperatures between 15°C and 30°C and moderate levels of humidity.
* **Autonomous Operation**: The robot should function autonomously, with minimal human intervention required for routine operations.
* **Compatibility**: The system should be compatible with standard power supplies and easy to integrate with existing infrastructures in childcare centres.

**5. Maintainability and Support Requirements**

* **Software Updates**: The system should support over-the-air software updates to ensure it remains up to date with the latest features and security patches.
* **Modularity**: The robot’s design should be modular, allowing easy replacement or upgrading of individual components without requiring complete system overhauls.
* **Technical Support**: Comprehensive technical support should be provided, including user manuals, online resources, and customer service.

**6. Security Requirements**

* **Data Encryption**: All data transmitted by the robot, particularly location and any personal data, should be encrypted using industry-standard protocols.
* **User Authentication**: Access to the robot’s settings and control systems should be restricted through robust user authentication mechanisms.
* **Tamper Resistance**: The robot should be physically tamper-resistant to prevent unauthorized access to internal components.

**7. Cultural and Political Requirements**

* **Cultural Sensitivity**: The robot should respect cultural norms in different regions, avoiding behaviours or designs that could be considered inappropriate or offensive.
* **Localization**: The robot should be customizable to adhere to local childcare regulations and practices.

**8. Legal Requirements**

* **Compliance with Safety Standards**: The robot must comply with relevant safety regulations, including child safety standards for electronic devices in childcare environments.
* **Privacy Compliance**: The system must adhere to data protection regulations, such as GDPR or equivalent local laws, ensuring that any data collected is handled with strict confidentiality.
* **Certification**: The robot should obtain necessary certifications for safe operation in childcare environments, such as CE marking or equivalent certifications.

**9. Reliability Requirements**

* **System Availability**: The robot should maintain an uptime of at least 99.9%, ensuring it is operational throughout the childcare centre’s hours.
* **Fault Tolerance**: The system should be able to handle minor hardware or software faults without complete failure, allowing for graceful degradation of functionality rather than total shutdown.
* **Backup and Recovery**: The system should have mechanisms for data backup and recovery in case of system crashes or power failures.

# System Navigation and Interface

## System navigation

Figure 2: System Navigation

## Interface

A screenshot of a computer

Description automatically generated

Figure 3: Virtual Machine

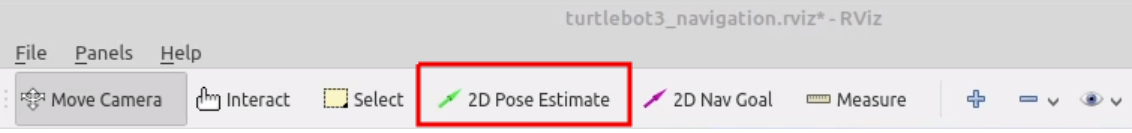
Oracle VM VirtualBox Manager is utilized to host the Ubuntu machine on laptops, enabling the operation and management of the robot's software via a virtual environment.

A screen shot of a computer program

Description automatically generated

Figure 4: Controls for Navigation

The robot can be easily controlled after it is setup properly. The commands above are used to control its movement.



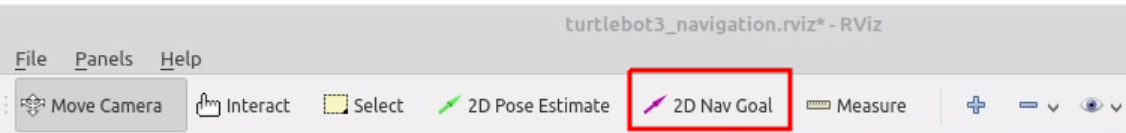


Figure 5: Rviz Controls

These two buttons in Rviz allow us to setup the Robot for navigation/guiding the child. The first one estimates the position of the robot and the second specifies the desired location it wants the robot to navigate to.

A screenshot of a computer program

Description automatically generated

Figure 6: Interface for Running Commands

The commands needed to run the nodes are almost always the same and require very few adjustments.

# Cost Estimate

Our estimated cost for the development services outlined above is $84,194.

|  |  |  |
| --- | --- | --- |
| **Category** | **Item** | **Cost** |
| **Hardware Costs** | TurtleBot3 Burger Robot | $500 |
|  | LiDAR Sensor (LDS-02) | $200 |
|  | Raspberry Pi | $50 |
|  | OpenCR (Microcontroller Unit) | $70 |
|  | IMU (Inertial Measurement Unit) | $30 |
|  | 12V Li-Po Battery | $40 |
|  | Miscellaneous Components | $100 |
|  | Spare Parts and Maintenance | $100 |
| *Total Hardware Costs* |  | **$1090** |
| **Software Costs** | Robot Operating System (ROS) | Free |
|  | Gazebo (simulation software) | Free |
|  | Additional Software Libraries and Tools | $100 |
| *Total Software Costs* |  | **$100** |
| **Personnel Costs** | Researchers (3 people, 20 hours/week each for 9 months at $30/hour) | $64,800 |
|  | Supervisor (1 person, 5 hours/week for 9 months at $50/hour) | $9,000 |
|  | Consultant (10 hours total at $100/hour) | $1,000 |
| *Total Personnel Costs* |  | **$74,800** |
| **Miscellaneous Costs** | Data Collection Expenses | $200 |
|  | Travel Expenses | $300 |
|  | Printing and Documentation | $50 |
| *Total Miscellaneous Costs* |  | **$550** |
| *Contingency* | Buffer for Unexpected Costs | **10% of Budget ($76,540) = $7,654** |
| *Total Project Budget* |  | **$84,194** |

Table 1: Cost and Budget

# Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **WBS Level** | **WBS Code** | **WBS Name** | **WBS Description** |
| **1** | **1** | **Childcare Robot Project** | Development and deployment of a 2D SLAM-based robot for enhancing safety and efficiency in childcare centers. |
| **2** | **1.1** | **Initiation** |  |
| 3 | 1.1.1 | Charter | Development of a charter that defines project goals, major stakeholders, and the approval process. |
| 3 | 1.1.2 | Stakeholder Setup | Identify stakeholders and establish communication protocols. |
| 3 | 1.1.3 | Feasibility & Risk | |  | | --- | |  |  |  | | --- | | Conduct feasibility study and risk assessment. | |
| 3 | 1.1.4 | Funding & Resources | Secure necessary funding and resources. |
| **2** | **1.2** | **Planning** |  |
| 3 | 1.2.1 | Project Plan | Develop detailed project plan with timelines, resources, and budget. |
| 3 | 1.2.2 | Requirements Spec | Finalize technical requirements for hardware and software. |
| 3 | 1.2.3 | Procurement Planning | Organize procurement of hardware and software. |
| 3 | 1.2.4 | Standards & Compliance | Establish protocols for quality, privacy, risk management, and safety. |
| **2** | **1.3** | **Execution** |  |
| 3 | 1.3.1 | Hardware & Software Setup | Assemble hardware and install software. |
| 3 | 1.3.2 | Testing & Integration | Conduct preliminary testing and system integration. |
| 3 | 1.3.3 | |  | | --- | |  |  |  | | --- | | Software Dev & Simulation | | Develop and simulate software functionalities. |
| 3 | 1.3.4 | Deployment & Evaluation | Deploy system in real-world settings and evaluate performance. |
| **2** | **1.4** | **Control** |  |
| 3 | 1.4.1 | Performance Reviews | Regularly review project performance and progress. |
| 3 | 1.4.2 | Quality & Risk Management | Manage project risks and maintain quality standards. |
| **2** | **1.5** | **Closure** |  |
| 3 | 1.5.1 | Feedback & Adjustments | Incorporate feedback and make final project adjustments. |
| 3 | 1.5.2 | Marketing & Outreach | Prepare and execute marketing materials. |
| 3 | 1.5.3 | Close | Document project outcomes and close the project. |

Table 2: Word Breakdown Structure

1.Childcare Robot Project

Figure 7: WBS Diagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task ID** | **Task Name** | **Duration** | **Start** | **Finish** | **Predecessors** |
| **1** | **Initiation Phase** | **28d** | **07 Aug '24** | **13 Sep '24** |  |
| 2 | Define project scope and goals. | 7d | 07 Aug '24 | 15 Aug '24 |  |
| 3 | Identify project stakeholders and establish communication protocols. | 7d | 16 Aug '24 | 26 Aug '24 | 2 |
| 4 | Conduct feasibility study and initial risk assessment. | 7d | 27 Aug '24 | 04 Sep '24 | 3 |
| 5 | Secure initial funding and resources. | 7d | 05 Sep '24 | 13 Sep '24 | 4 |
| **6** | **Planning Phase** | **42d** | **16 Sep '24** | **12 Nov '24** |  |
| 7 | Develop detailed project plan | 14d | 16 Sep '24 | 03 Oct '24 | 5 |
| 8 | Finalize requirements for hardware and software components | 7d | 04 Oct '24 | 14 Oct '24 | 7 |
| 9 | Plan procurement processes for hardware and software | 7d | 15 Oct '24 | 23 Oct '24 | 8 |
| 10 | Establish quality control, privacy standards, risk management and safety certification protocols | 14d | 24 Oct '24 | 12 Nov '24 | 9 |
| **11** | **Execution Phase** | **98d** | **24 Oct '24** | **20 May '25** |  |
| 12 | Hardware Setup and Software Configuration | 21d | 24 Oct '24 | 21 Nov '24 | 9 |
| 13 | Preliminary Testing and Integration | 7d | 22 Nov '24 | 02 Dec '24 | 12 |
| 14 | Software Development and Simulation | 42d | 03 Dec '24 | 10 Apr '25 | 13 |
| 15 | Real-World Deployment and Evaluation | 28d | 11 Apr '25 | 20 May '25 | 14 |
| **16** | **Control Phase** | **126d** | **16 Sep '24** | **10 Mar '25** |  |
| 17 | Regular Performance and Progress Reviews | 126d | 16 Sep '24 | 10 Mar '25 | 5 |
| 18 | Quality Control and Risk Management | 77d | 22 Nov '24 | 10 Mar '25 | 12 |
| **19** | **Closing Phase** | **28d** | **21 May '25** | **27 Jun '25** |  |
| 20 | Final adjustments based on feedback and compliance results | 7d | 21 May '25 | 29 May '25 | 15 |
| 21 | Preparation and execution of marketing and outreach materials | 14d | 30 May '25 | 18 Jun '25 | 20 |
| 22 | Project review and closure documentation | 7d | 19 Jun '25 | 27 Jun '25 | 21 |

Table 3: Activities List

A screenshot of a computer

Description automatically generated

Figure 8: Gantt Chart

**Schedule Overview**

* **Initiation Phase (28 days):** This phase sets the foundation for the project. Tasks include defining the project scope, identifying stakeholders, conducting a feasibility study, and securing necessary funding and resources. This phase is crucial as it establishes the project's parameters and ensures that all necessary support and approvals are in place before moving forward.
* **Planning Phase (42 days):** Detailed planning involves developing the project plan, finalizing hardware and software requirements, planning procurement, and establishing standards for quality control, privacy, risk management, and safety. This phase prepares the project for successful execution by laying down a detailed roadmap and guidelines that will govern project activities.
* **Execution Phase (98 days):** The core phase where actual product development happens, including hardware setup and software configuration, preliminary testing and integration, software development and simulation, and real-world deployment and evaluation. This phase is critical as it transforms theoretical plans and designs into a tangible, operational system.
* **Control Phase (126 days):** This phase overlaps substantially with others due to its nature, focusing on monitoring the project's progress and maintaining standards through regular performance reviews and risk management activities. It ensures the project remains on track and meets set quality standards.
* **Closing Phase (28 days):** The final phase involves making adjustments based on feedback, preparing and executing marketing and outreach materials, and completing project documentation for closure. This phase wraps up the project by ensuring all objectives have been met and the product is ready for wider deployment and use.

**How These Relate to the Project**

1. **Sequential and Overlapping Tasks:** The schedule shows dependencies where certain tasks cannot begin until previous ones are completed, such as "Hardware Setup and Software Configuration" that can't start until procurement plans are finalized. Meanwhile, the control activities like "Regular Performance and Progress Reviews" span multiple phases, emphasizing continuous oversight.
2. **Critical Path:** Identifying tasks that directly affect the project timeline helps in prioritizing efforts and resources. For instance, delays in "Software Development and Simulation" would directly delay "Real-World Deployment and Evaluation," impacting the project's finish date.
3. **Resource Allocation:** Knowing the duration of each phase helps in planning resource allocation efficiently, ensuring that manpower and financial resources are available when needed without idling or bottlenecks.
4. **Risk Management:** Scheduling long durations for tasks like risk management and quality control highlights the project’s commitment to adhering to standards and addressing potential issues proactively.
5. **Stakeholder Engagement:** Scheduled reviews and documentation phases ensure continuous engagement with stakeholders, keeping them informed and involved throughout the project lifecycle, which is crucial for maintaining alignment and support.

# Considerations

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Description** | **Possible Solution** | **Justification** |
| Technical Risk | Building and deploying the robot comes with some technical challenges, like making sure it can accurately map and move around different daycare environments. Problems could pop up with the robot’s sensors, software, or its ability to work in tricky spaces with complex layouts or shiny surfaces. | To avoid these issues, we’ll do thorough testing in a variety of real-world-like settings. This includes testing in different lighting, with obstacles, and on different floor types. Keeping the software modular also makes updates and fixes easier. | Catching and fixing these problems early during testing means we’re less likely to face major issues later, ensuring the robot works well in any daycare setting. |
| Privacy Risk | Even though we’re using LiDAR sensors instead of cameras to protect privacy, there’s still a chance that data could be mishandled or accessed by the wrong people. | We’ll use data encryption to make sure any data the robot collects is secure. | Protecting data privacy helps build trust with users, which is key to getting the robot widely accepted in sensitive places like daycare centers. |
| Transitioning to New System | Switching from traditional child supervision methods to a robotic system might be disruptive. Staff might struggle to adapt to the new technology, and there could be worries about job security or how reliable the robot is. | We’ll roll out the system gradually, starting with a pilot program in a few daycare centers. We’ll provide thorough training and support, including hands-on demos and easy-to-follow manuals. The rollout will expand based on feedback and any needed adjustments. | A phased rollout helps ensure a smooth transition, making it easier for staff to get comfortable with the robot. This reduces the risk of disruption and increases the chances of success. |
| Over Budget | The project might go over budget due to unexpected costs, like needing extra hardware, longer development time, or unforeseen technical issues. | We’ll keep a close eye on the budget and put strict controls in place. We’ll set aside a contingency fund for any surprises and focus spending on the most critical parts of the project. Regular budget reviews will help keep things on track. | By staying on top of the budget and having a backup plan, we can avoid running out of money and make sure we deliver the essential features without cutting corners. This also helps maintain trust with stakeholders. |

Table 4: Consideration and Issues

# Human Factors and Privacy Preserving

When developing the 2D SLAM-based childcare robot, it’s crucial that the system is easy for daycare staff to use with minimal training. Our goal is a user-friendly design that fits smoothly into daily routines.

A key challenge is ensuring the robot’s interface isn’t too complex, which could lead to frustration and mistakes. To avoid this, we’re focusing on a simple, clear design with easy-to-recognize icons and prompts, making the robot intuitive to use. While the design will be straightforward, some training will still be provided, including tutorials and hands-on sessions to ensure everyone is comfortable with the robot. We’ll also set up a feedback system so staff can share their experiences, helping us make improvements based on real-world use. Prototyping and user testing will help catch any usability issues early, and simplified controls will make the robot easy for all staff to operate. After deployment, ongoing feedback will guide further refinements to keep the system responsive to daycare needs.

Privacy is also a top priority. The robot uses LiDAR sensors instead of cameras to avoid capturing personal images, greatly reducing privacy risks. We’ll implement strong data protection measures, like encryption, to ensure that any collected data is securely stored and only accessible to authorized personnel.

By prioritizing ease of use, intuitive design, and privacy protection, we aim to develop a childcare robot that seamlessly integrates into daycare operations, is easy to adopt, effective in its role, and trusted by all users.

# Risk and Issue mitigation

Our project involves a 2D SLAM-based robot designed for childcare, focusing on safety and navigation without cameras, using LiDAR sensors. Despite not capturing images, privacy risks arise from tracking movements, addressed by encrypting data and ensuring transparency about the robot's function to alleviate concerns of constant surveillance. Local data storage policies will emphasize secure deletion after specific periods, aligning with best privacy practices without seeking certifications. Key risks include moderate privacy concerns from data collection (Risk Level: 9), low risk of perceived surveillance (Risk Level: 4), high risk related to data management (Risk Level: 12), and compliance practices (Risk Level: 6).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Risk Factor** | **Likelihood (1-5)** | **Impact (1-5)** | **Risk Level (Likelihood x Impact)** | **Description** |
| 1 | Privacy Concerns from LiDAR Data Collection | 3 (Moderate) | 3 (Significant) | 9 (Moderate) | Privacy issues from data collection could arise, needing strong encryption and policies. |
| 2 | Perception of Constant Surveillance | 2 (Low) | 2 (Minor) | 4 (Low) | Staff or parents may feel overly monitored; transparency and demonstrations can mitigate this. |
| 3 | Data Management and Storage | 3 (Moderate) | 4 (Major) | 12 (High) | Improper data management could lead to significant privacy breaches; robust data policies required. |
| 4 | Compliance and Best Practices | 2 (Low) | 3 (Medium) | 6 (Low) | Ensuring adherence to privacy standards and best practices through regular review, even without specific certifications. |

Table 5: Risk Matrix

# Ethical Issues

Creating a childcare robot is exciting, but it comes with important ethical questions. First, we need to protect privacy. The robot will collect some data, so we’ll make sure it’s encrypted and let parents and staff know how it’s used and stored. Being open about this builds trust.

It’s also important that everyone understands and agrees to the robot being in the daycare. We’ll clearly explain what the robot does and how it helps keep kids safe, so everyone feels comfortable with it. We need to be honest about what the robot does and how it benefits the children. Transparency builds trust and reassures everyone that the robot is there to help, not to replace caregivers. If something goes wrong, like a malfunction, we’ll have clear roles and responsibilities to address issues quickly. And while some might worry about job security, the robot is designed to assist, not replace, human caregivers.

# References

1. Lam, T. (2020). *The 7 most common childhood injuries | HealthPartners Blog*. [online] HealthPartners Blog. Available at: https://www.healthpartners.com/blog/most-common-childhood-injuries.
2. ‌QR Research (2023). *Market Overview and Report Coverage Childcare robots are robotic devices designed to assist parents and caregivers in taking care of children. These robots are equipped with various features such as remote monitoring capabilities, interactive play, educational content, and basic caregiving functions*. [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/childcare-robots-market-challenges-opportunities-growth-drivers-fi8qe/.
3. Name, Y. (n.d.). *ROBOTIS e-Manual*. [online] ROBOTIS e-Manual. Available at: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/.
4. ‌Open Robotics (2020). *ROS.org | Powering the world’s robots*. [online] Ros.org. Available at: https://www.ros.org/.
5. ‌Tee, Y.K. and Han, Y.C. (2021). *Lidar-Based 2D SLAM for Mobile Robot in an Indoor Environment: A Review*. [online] IEEE Xplore. doi:https://doi.org/10.1109/GECOST52368.2021.9538731.
6. ‌Lai, J., Lei, H., Bao, S., Du, L., Yuan, J. and Ma, S. (2021). Design of a Portable Indoor Guide Robot for Blind People. doi:https://doi.org/10.1109/icarm52023.2021.9536077.
7. ‌Takleh Omar Takleh, T., Abu Bakar, N., Abdul Rahman, S., Hamzah, R. and Abd Aziz, Z. (2018). A Brief Survey on SLAM Methods in Autonomous Vehicle. *International Journal of Engineering & Technology*, 7(4.27), p.38. doi:https://doi.org/10.14419/ijet.v7i4.27.22477.
8. ‌illusion, S. (2023). *Localization, Mapping and SLAM using GMapping*. [online] Medium. Available at: https://softillusion-robotics.medium.com/gmapping-d26c13b1b69.
9. ‌Guivant, J.E. and Nebot, E.M. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3), pp.242–257. doi:https://doi.org/10.1109/70.938382.
10. ‌ Zhang, Y., Xiao, Z., Yuan, X., Li, S. and Liang, S. (2019). Obstacle Avoidance of Two-Wheeled Mobile Robot based on DWA Algorithm. doi:https://doi.org/10.1109/cac48633.2019.8996425.
11. Zubaidah Al-Mashhadani, Manasa Mainampati and Chandrasekaran, B. (2020). Autonomous Exploring Map and Navigation for an Agricultural Robot. doi:https://doi.org/10.1109/iccr51572.2020.9344404.
12. ‌ Josep Aulinas, Yvan Petillot, Salvi, J. and Lladó, X. (2008). The SLAM problem: a survey. pp.363–371. doi:https://doi.org/10.3233/978-1-58603-925-7-363.
13. ‌Zhang, G.L., Yao, E.L., Tang, W.J. and Xu, J. (2014). An Improved Particle Filter SLAM Algorithm in Similar Environments. *Applied Mechanics and Materials*, 590, pp.677–682. doi:https://doi.org/10.4028/www.scientific.net/amm.590.677.

# Appendix

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Time** | **Items to be discussed** | **Attended** |
| 10/08/2024 | 3:00 PM | **Agree on Common Availability:** Establish a schedule that suits all team members for meetings and work sessions. | Everyone |
| 14/08/2024 | 3:00 PM | **Reach Consensus on Project Objectives:** Discuss and finalize mutual understanding and agreement on the key aspects of the project.  **Divide Responsibilities:** Allocate tasks among team members based on expertise and workload balance. | Everyone |
| 15/08/2024 | 2:30 PM | **Consult with the Supervisor:** Address any uncertainties or seek guidance on project directions by consulting with the project supervisor.  **Progress on Individual Tasks:** Ensure each member advances their respective parts of the project as planned. | Everyone |
| 18/08/2024 | 12:00 PM | **Assemble and Review the Final Document:** Combine individual contributions into a comprehensive final document and conduct a thorough review for consistency and completeness. | Everyone |

|  |  |  |
| --- | --- | --- |
| **Members Name** | **Sections Written** | **Tasks Undertaken** |
| Mary | - Functional Requirements  - Meeting Minutes  - System navigation and UI  - Project Schedule  - Risks | - Distribution of work  - Writeup for the mentioned sections  -Setup meeting & maintain group register  - Helping and guiding team members with the topics |
| Meghna | - System Development Approach  - Use Case Diagram  - Member Contribution Table  - Non- Functional Requirements  - Risks | - Distribution of work  - Writeup for the mentioned sections  - Research case study |
| John | - Project Cost Estimate  - Issues and Solutions  - Human Factors  - Ethical Issues  - Risks | - Writeup for the mentioned sections  - Distribution of work  - Research case study |

Github Repository:

https://github.com/maryyaz/NIT6150