Humanoid Robot System Design

Introduction

Humanoid robots, engineered to mirror human capabilities, blend intricate software and hardware systems. This design document delves into the design of a Python-based humanoid robot system, outlining three primary functionalities. The design rationale, key system selections, and supporting references are also explored.

System Overview

The system comprises three features:

1. To-Do List: Implements task management where users can add, remove, and view tasks.
2. Stack Calculator: Utilises a stack for performing calculations with undo functionality.
3. Support Queue: Manages a support system queue, allowing users to add, serve, and monitor queued items.

Rationale

Robots are increasingly aiding humans in tasks once considered exclusively human, fostering human-robot collaboration. Ackerman (2023) underscores the practical application of humanoid robots in workplaces, emphasising efficiency and adaptability. As this field progresses, human-robot interactions evolve (Mukherjee et al., 2022).

Mulko (2023) notes that modern humanoid robots, driven by AI and machine learning, can imitate human behaviour and emotions. These robots find practical use in personal assistance, manufacturing, research, and rescue operations.

Patil and Patil (2024) highlight Python's simplicity, versatility, and widespread use in data science, software development, and web development. Its ease of learning, extensive libraries, and strong community support make it suitable for rapid prototyping and advanced algorithm integration. By encapsulating each feature within a distinct class, the system prioritises maintainability through high cohesion and low coupling.

The decision to structure the system using independent classes for each feature aligns with the modularity principle. Modularity simplifies debugging, promotes scalability, and enables independent updates to each module. Tilley (2024) emphasises the significance of modular systems in enhancing humanoid robot functionality and adaptability. Modularity in robotic design facilitates incremental improvements and customisation.

The selection of lists, stacks, and queues aligns with their inherent properties:

- Lists allow dynamic resizing and random access, ideal for managing to-do tasks.
- Stacks support Last-In-First-Out (LIFO) operations, making them suitable for implementing undo functionality.
- Queues adhere to First-In-First-Out (FIFO) principles, ensuring fair order processing in the support system.

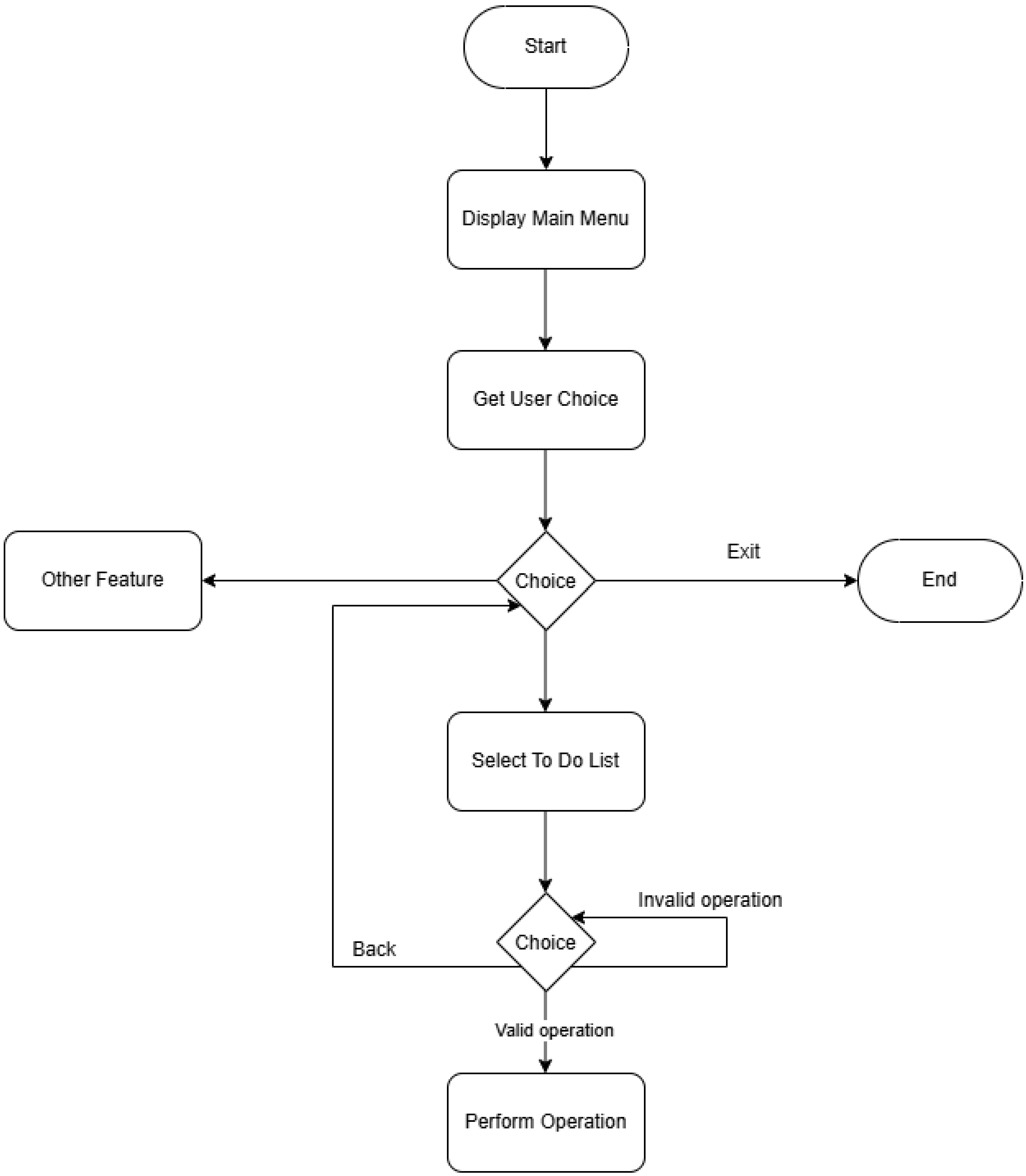
This design ensures efficient task handling and resource allocation, drawing on task scheduling algorithms commonly used in robotics (Russell and Norvig, 2022). Queues, particularly effective for managing prioritised and equitable processing, are well-suited for task or request management in systems like support management. Joo et al. (2022) highlight the use of FIFO-based queues in ensuring fairness and efficiency in multi-agent systems, a principle that enhances the operational robustness of robotic support modules.

UML Models

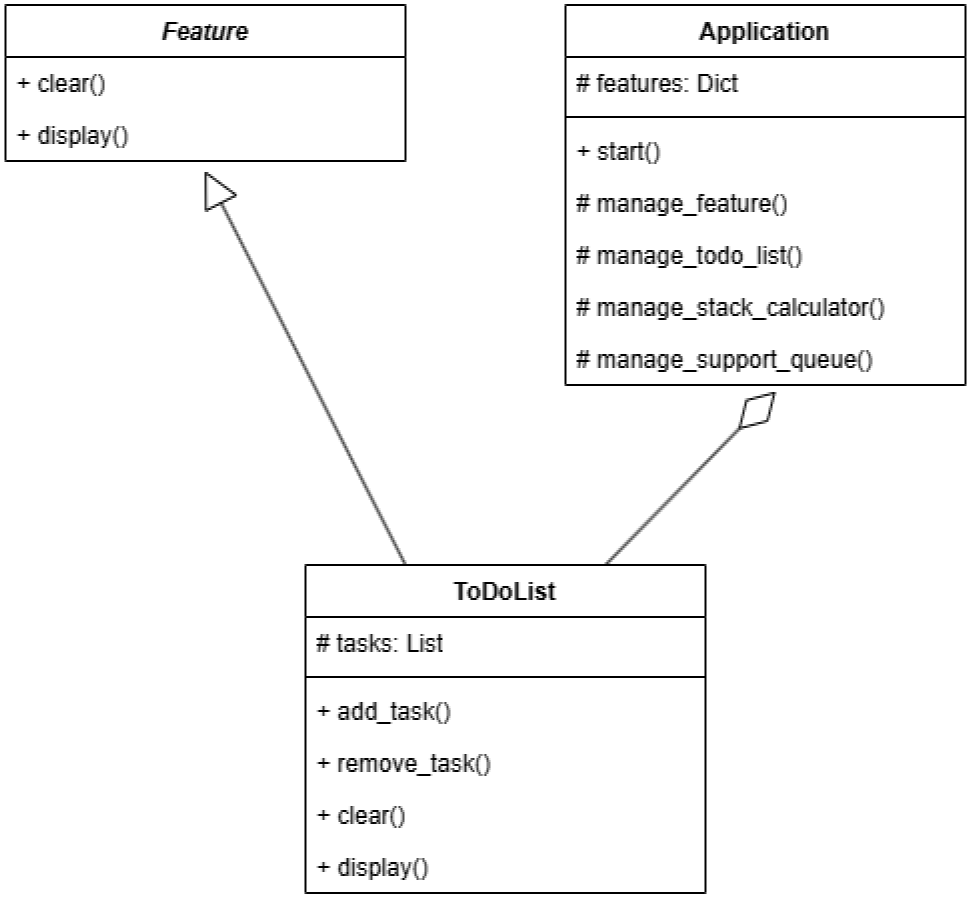
The following UML diagrams provide a visual representation of each feature, offering insights into the system's design, operations, and behaviour.

To-Do List

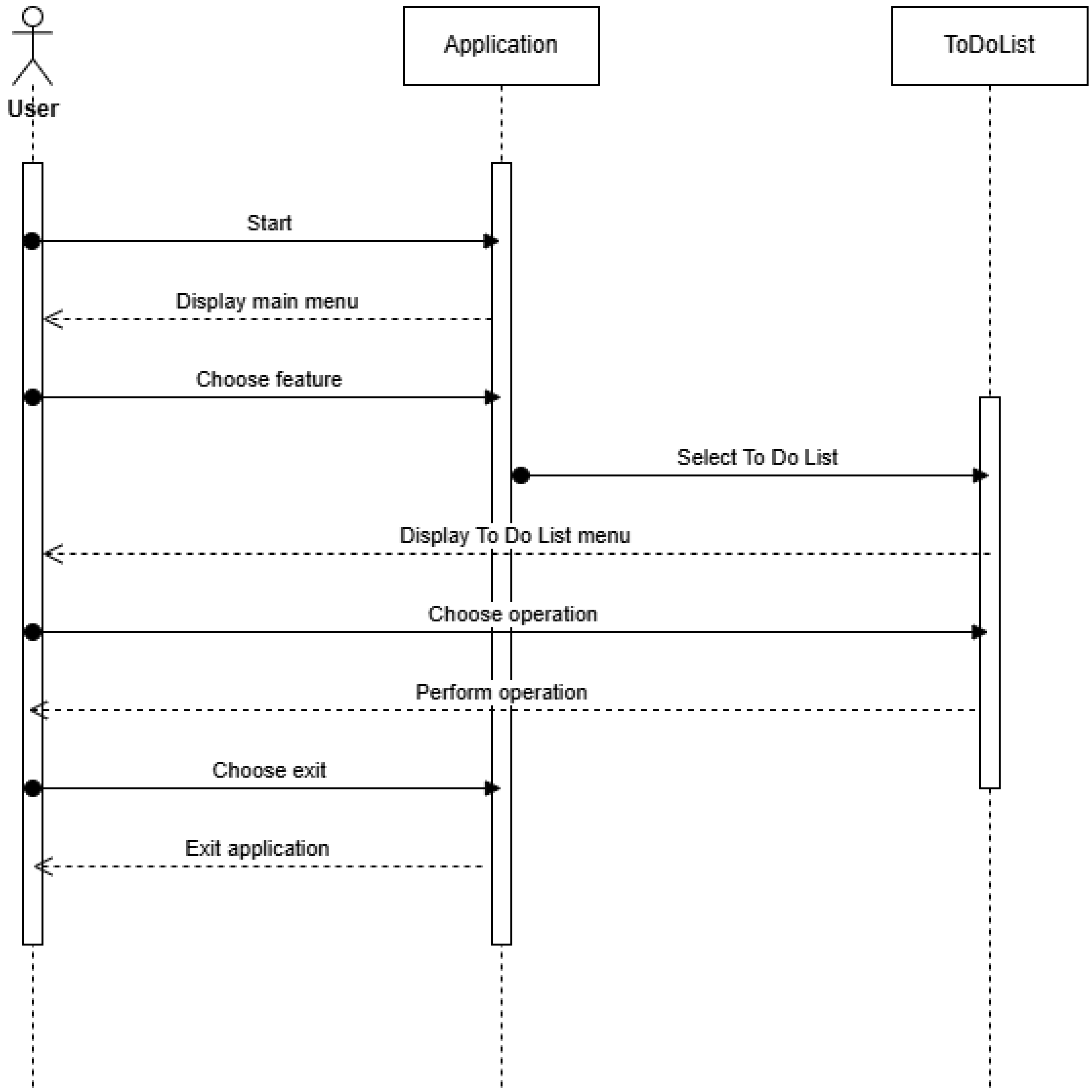
Activity diagram:



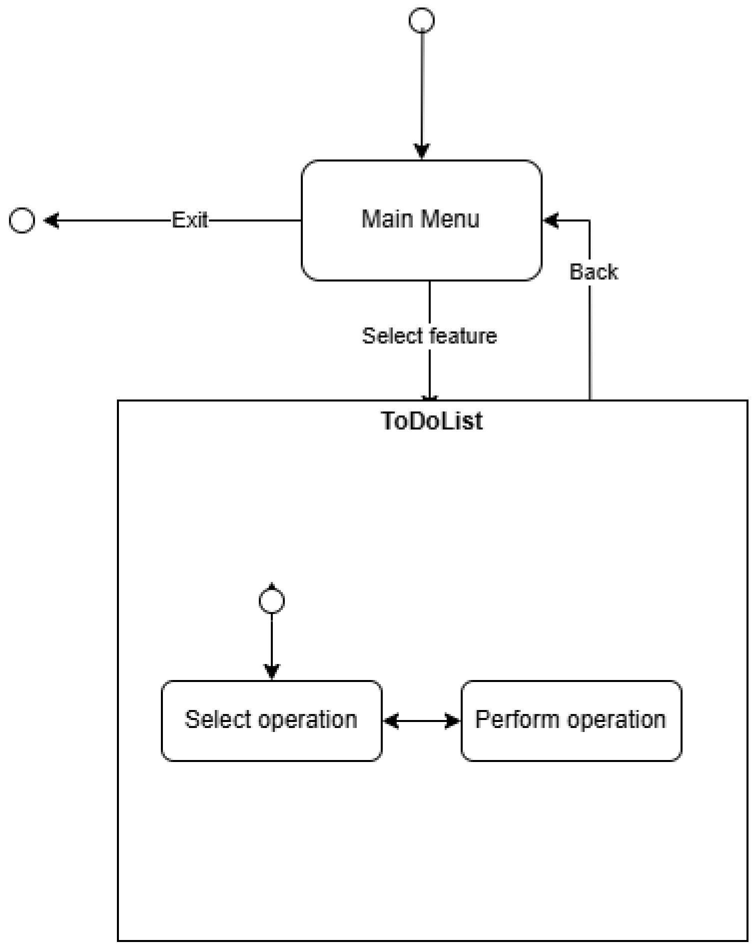
Class diagram:



Sequence diagram:

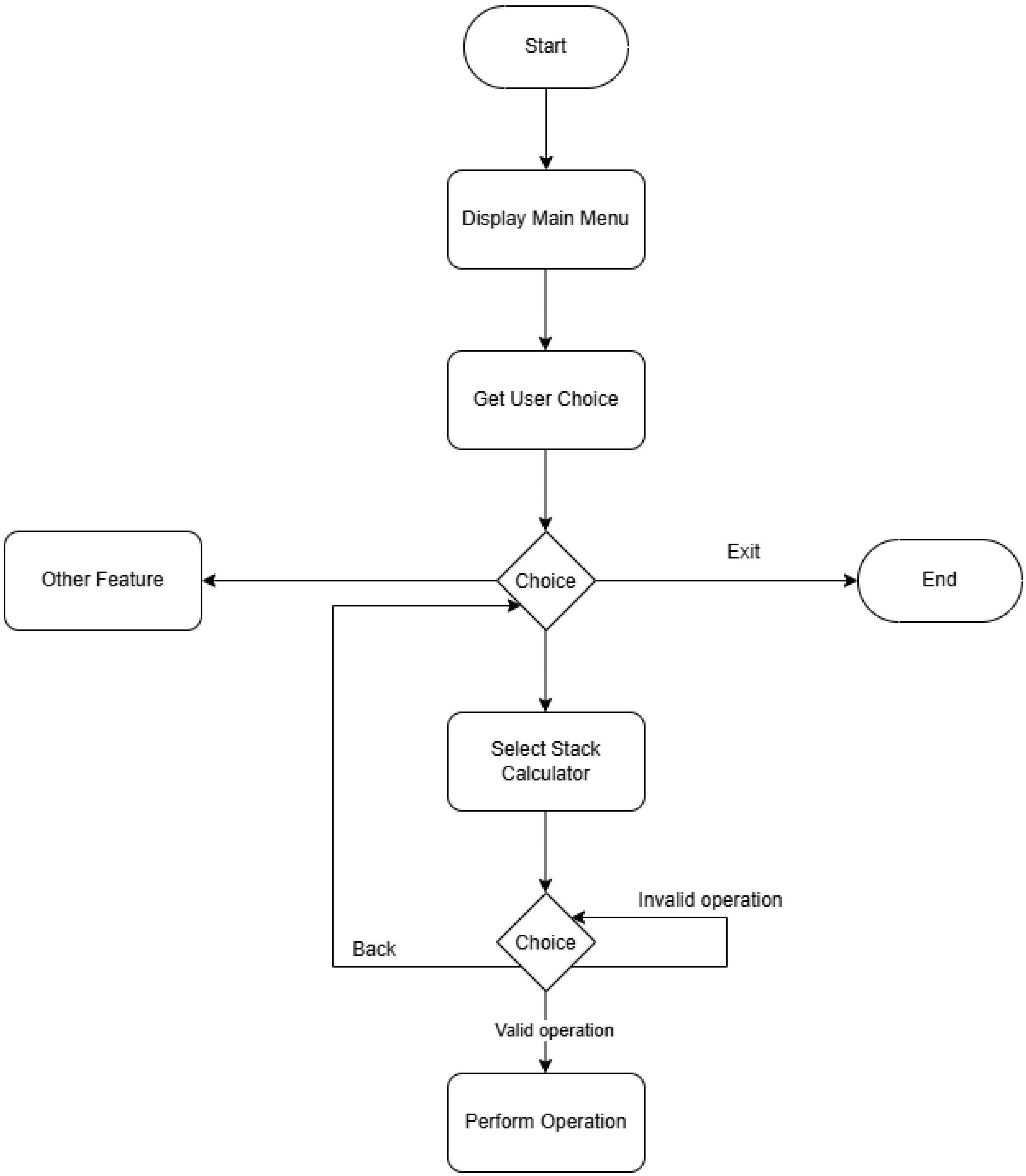


State transition diagram:

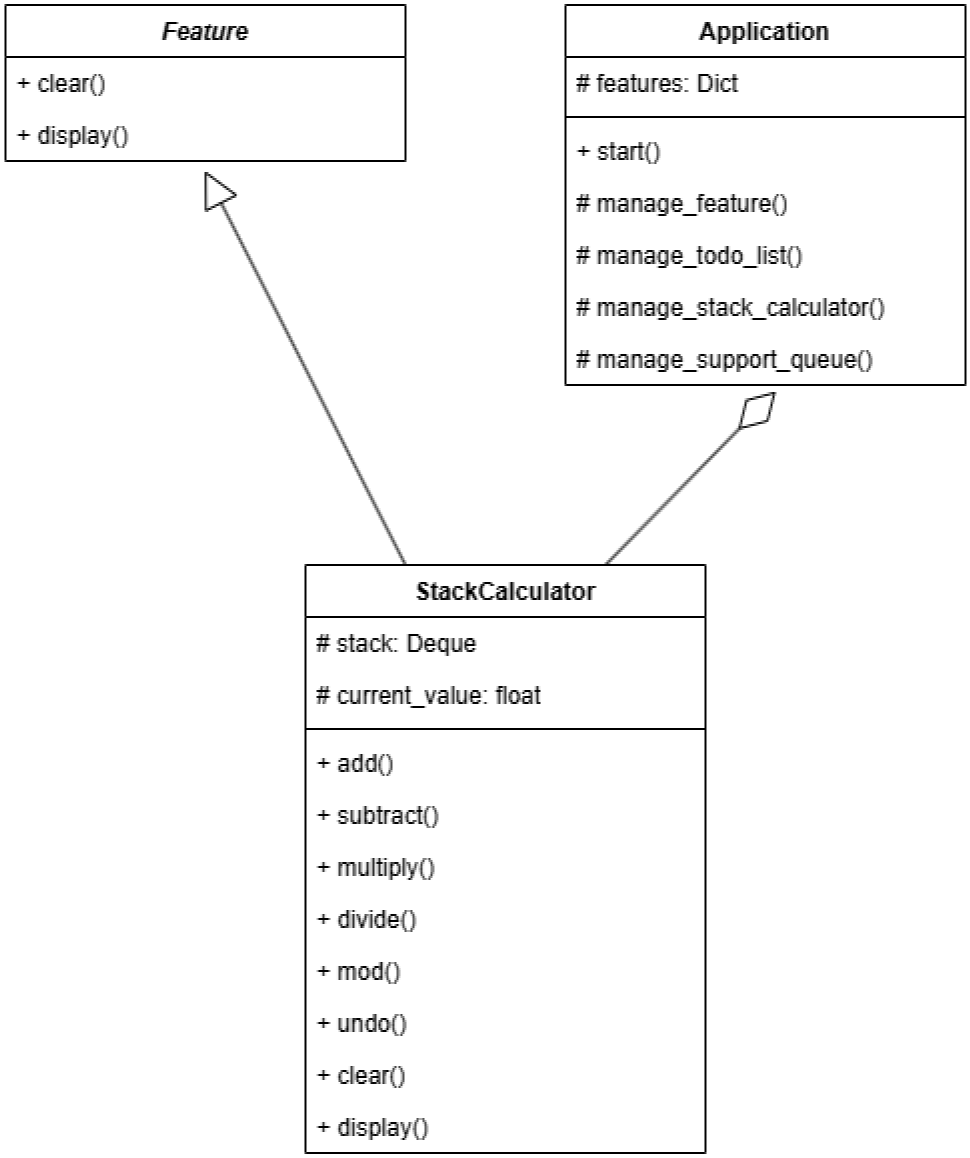


Stack Calculator

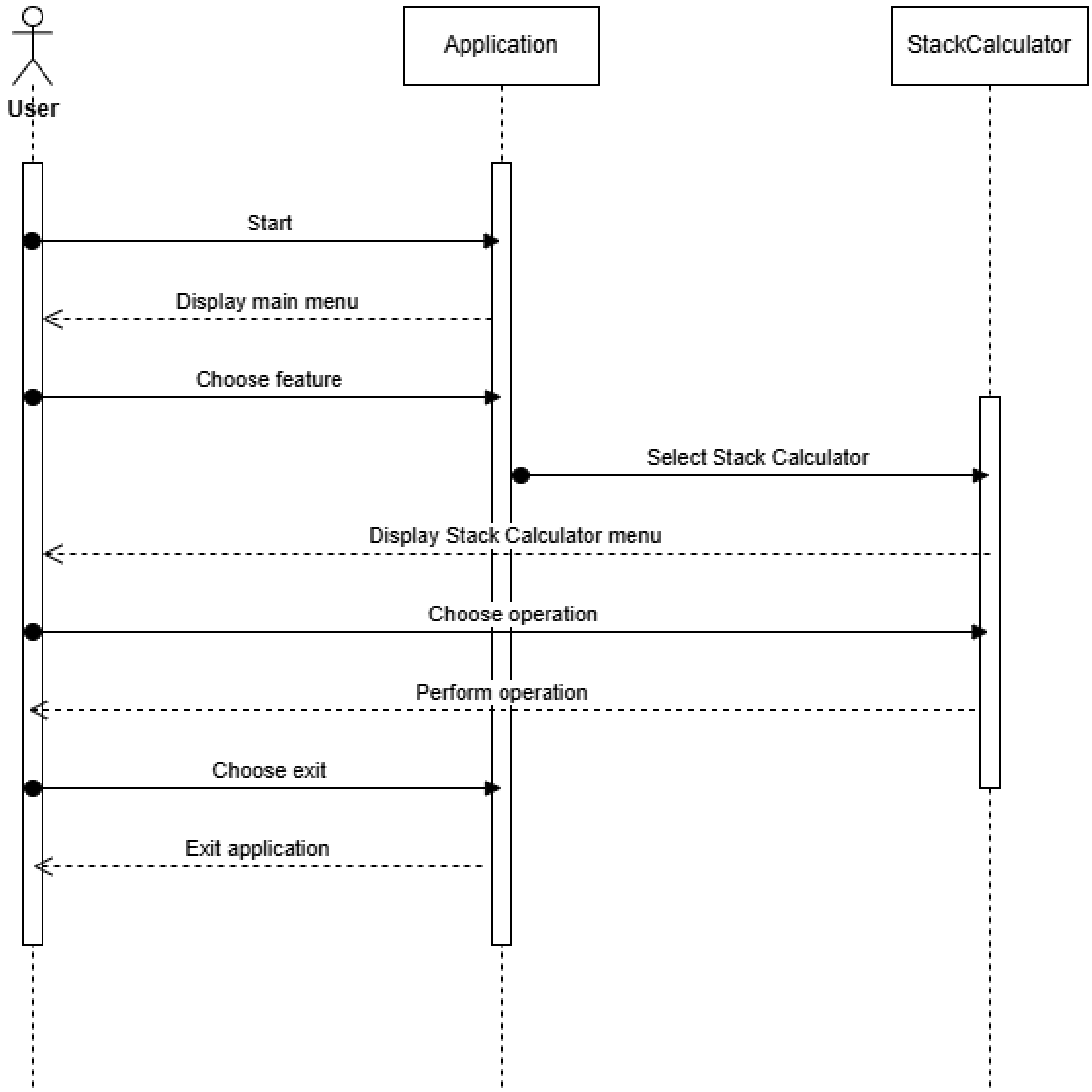
Activity diagram:



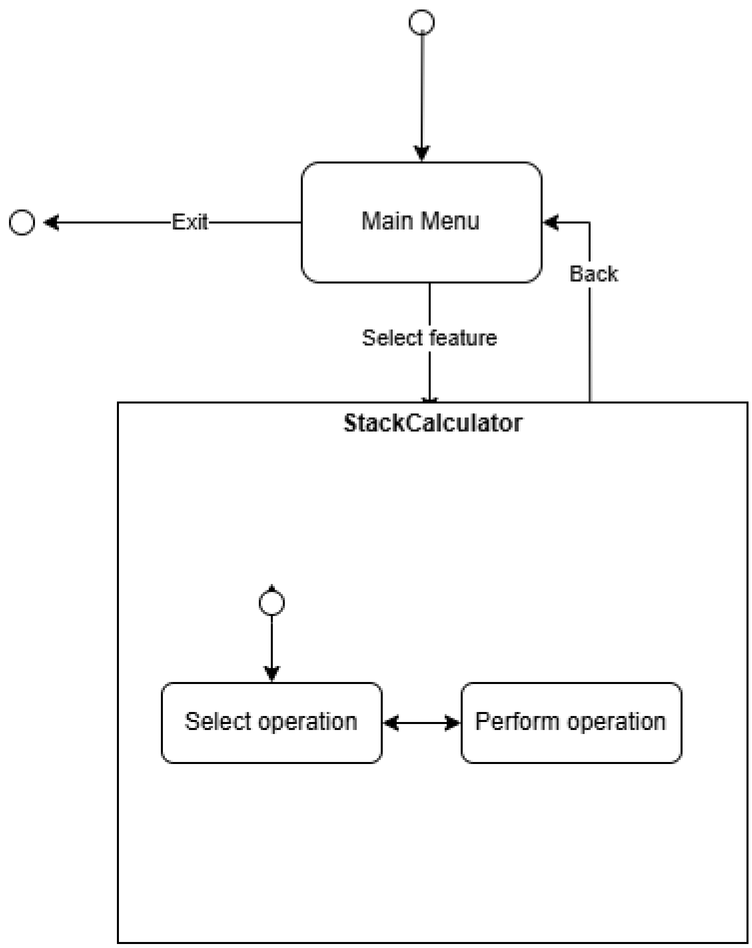
Class diagram:



Sequence diagram:

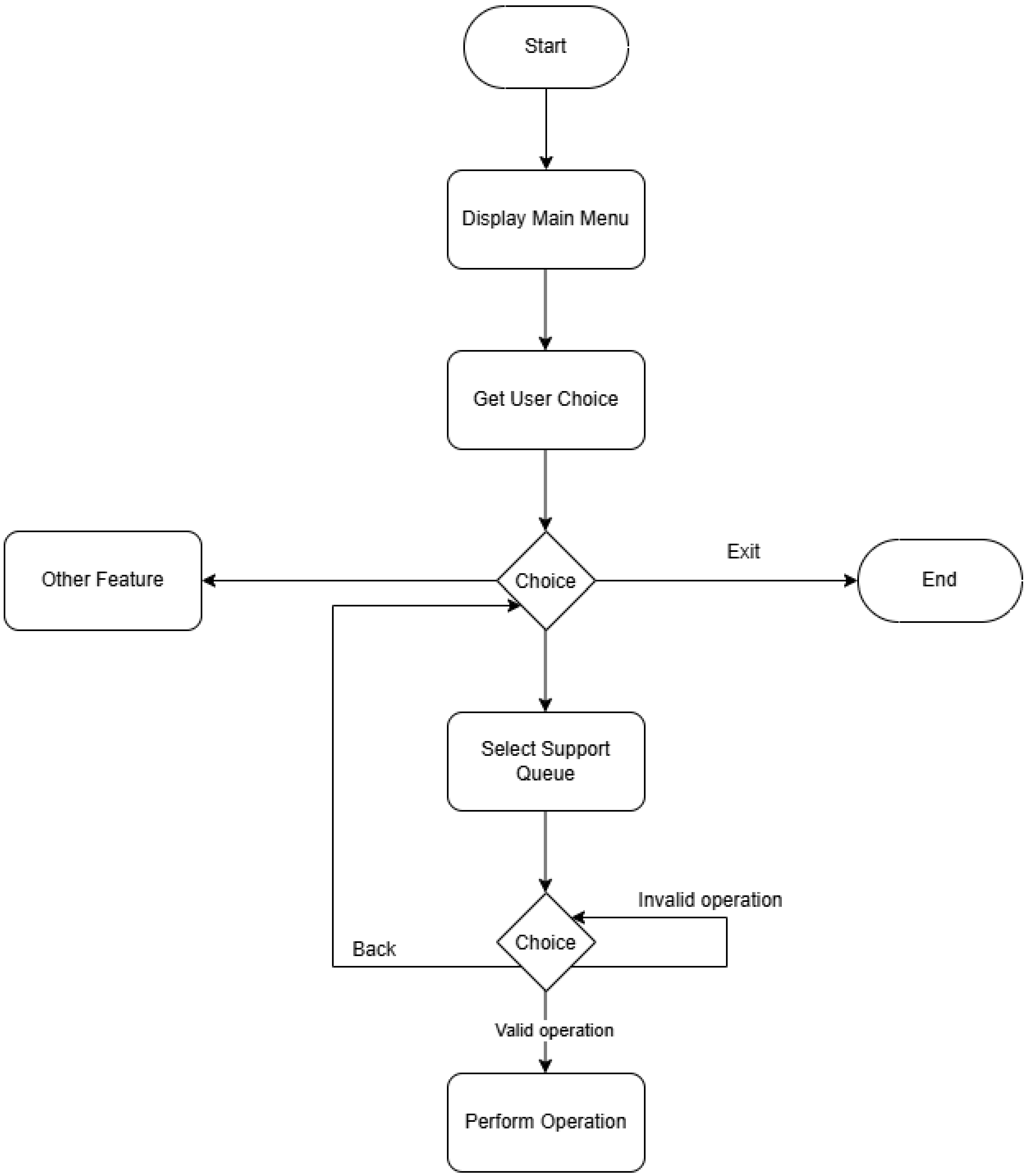


State transition diagram:

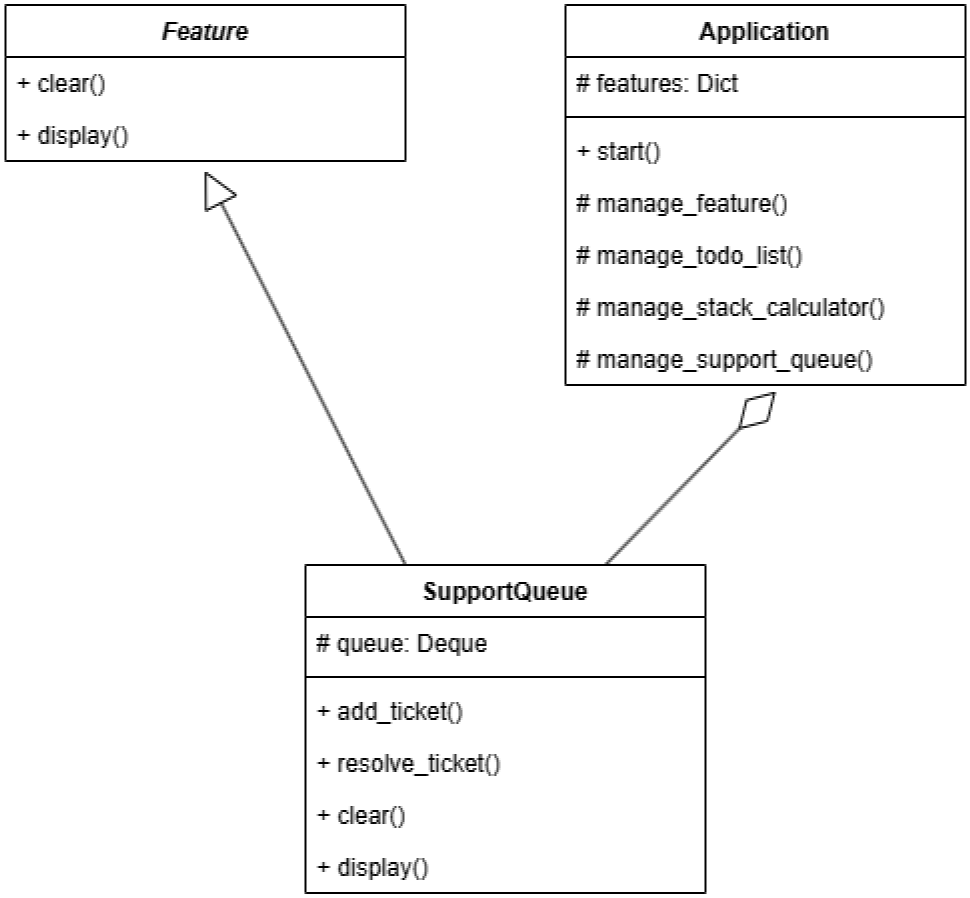


Support Queue

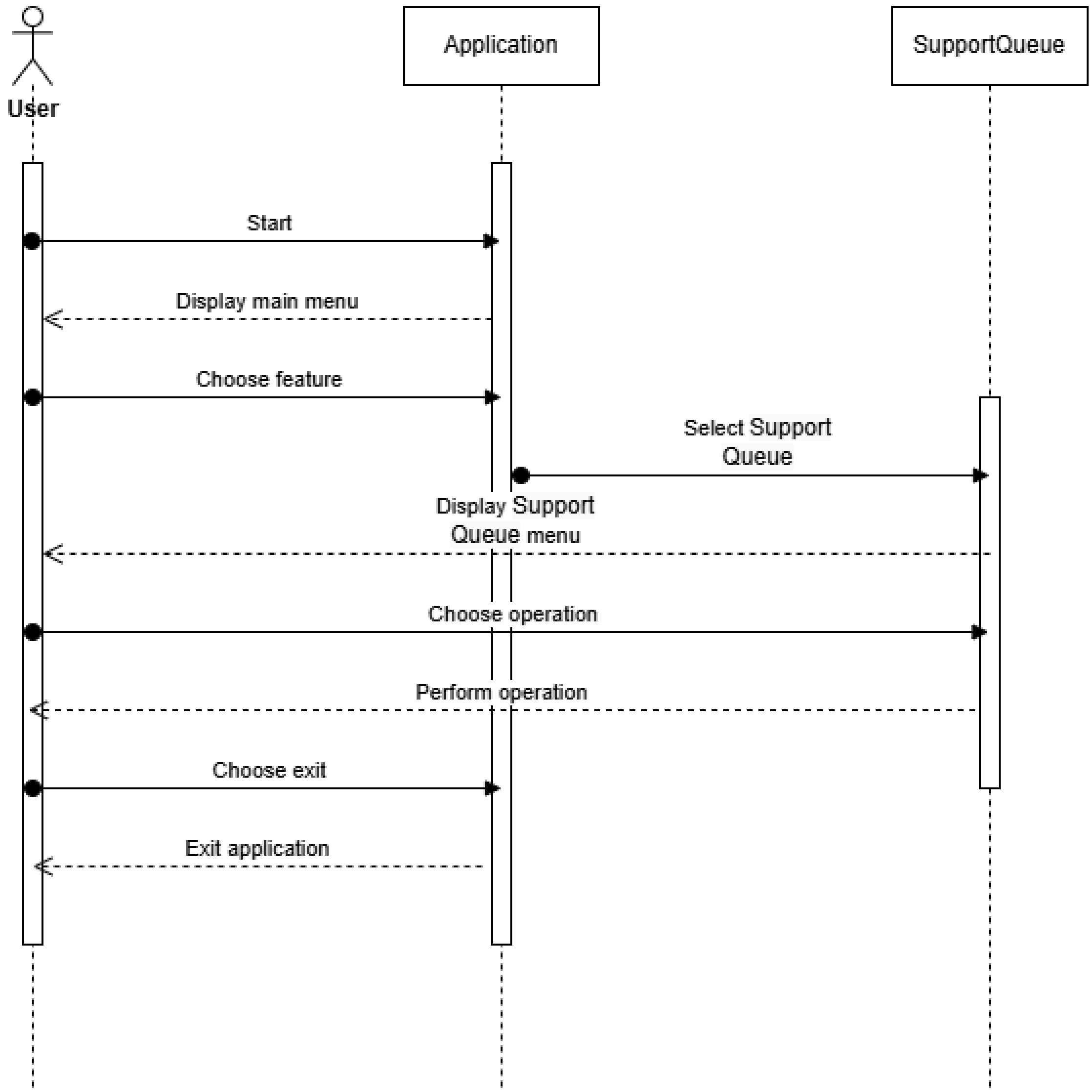
Activity diagram:



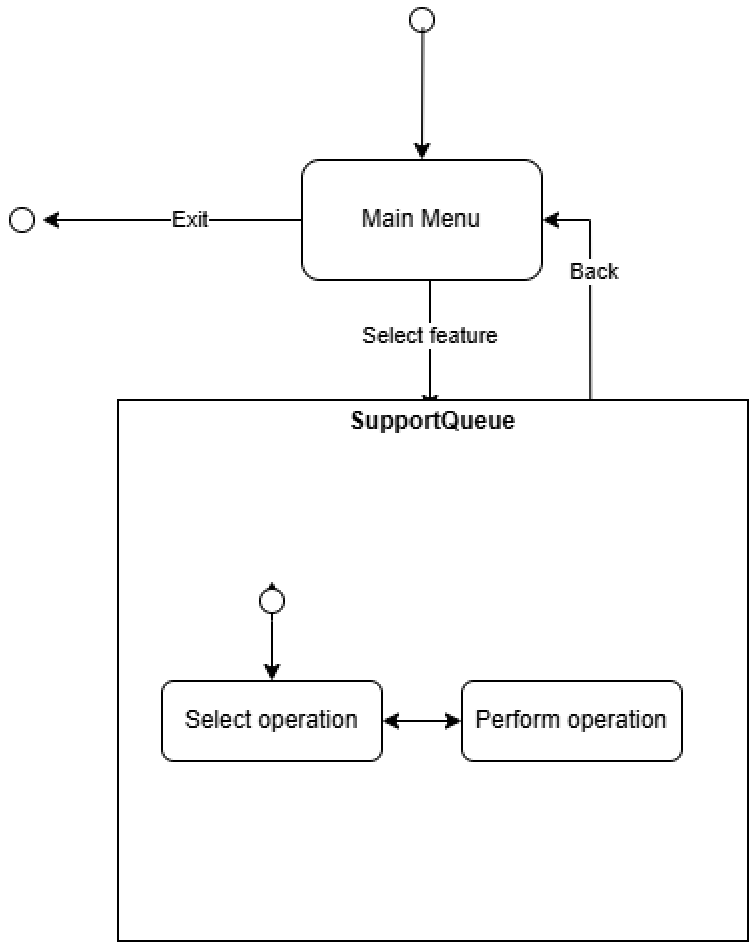
Class diagram:



Sequence diagram:



State transition diagram:



Conclusion

The humanoid robot system demonstrates the strength of modular design, utilising lists, stacks, and queues to implement user-orientated features. Reinforced by UML models, this design guarantees scalability, maintainability, and efficient task execution. The system offers a strong foundation for incorporating advanced capabilities, aligning with contemporary robotics principles and practices.

References

Ackerman, E. (2023) Humanoid Robots Are Getting to Work. *IEEE Spectrum*.

Joo, T., Jun, H & Shin, D. (2022) Task Allocation in Human–Machine Manufacturing Systems Using Deep Reinforcement Learning. *Sustainability* 14(4): 2245. DOI: https://doi.org/10.3390/su14042245

Mukherjee, D., Gupta, K., Chang, L.H. & Najjaran, H. (2022) A Survey of Robot Learning Strategies for Human-Robot Collaboration in Industrial Settings. *Robotics and computer-integrated manufacturing*, 73, 102231-.

Mulko, M. (2023) 5 of the world's most realistic humanoid robots ever. *Interesting Engineering*.

Russell, S. & Norvig, P. (2022) *Artificial Intelligence: A Modern Approach*. 4th ed. Harlow: Pearson Education Limited.

Patil S. & Patil S. (2024) A Review on Python the Fastest Growing Programming Language. *International Journal For Multidisciplinary Research* 6(3). DOI: https://doi.org/10.36948/ijfmr.2024.v06i03.23112

Tilley S. (2024) *Systems Analysis and Design*. 13th ed. Maon: Cengage.