Autonomous Office Organization Robot: System Design and Implementation Report

PDE3802 - Artificial Intelligence (AI) in Robotics

Academic Year 2025-26

Submitted by:

Oluwatunmise Raphael Shuaibu

Student ID: M00960413

MIDDLESEX UNIVERSITY

Submission Date: 7th November 2025

Contents

Section 1: Executive Summary

Modern office environments frequently suffer from workspace clutter as everyday items accumulate on desks, reducing efficiency and creating visual obstructions. This report presents the design of an autonomous office organization robot that employs artificial intelligence, computer vision, and robotic manipulation to detect, classify, and relocate common office items, thereby restoring and maintaining organized workspaces.

The system integrates a mobile manipulator platform with a robotic arm and depth camera for perception. It autonomously recognizes ten common office objects—including mugs, bottles, keyboards, phones, and stationery—using deep learning-based computer vision. The robot navigates safely through office spaces, plans grasping motions, and relocates items to designated storage areas, with built-in safety mechanisms including emergency stop controls and collision avoidance.

The proposed design targets ≥85% object recognition accuracy and ≥80% grasp success rate while maintaining a modular, cost-effective architecture based on the ROS 2 framework. Using off-the-shelf components, the system demonstrates practical feasibility for deployment in typical office environments, offering a scalable foundation for intelligent workspace automation.

Section 2: Introduction & Mission Objectives

Modern office environments often become cluttered as workers accumulate everyday items such as mugs, bottles, stationery, and electronic devices. This clutter not only reduces workspace efficiency but can also create physical and visual obstructions. Advances in artificial intelligence (AI) and robotics enable the automation of such repetitive organizational tasks, supporting cleaner and more efficient office spaces.

This project proposes the design of an autonomous office-desk organizing robot capable of detecting, classifying, and repositioning common office items within an indoor workspace. The system integrates computer vision, perception-driven planning, and safe manipulation to restore desk order without continuous human supervision. It performs organized, discrete missions, each beginning with a start command and ending when the workspace meets defined organization criteria. The robot is designed to operate in typical indoor office settings, handling common desk-level objects while maintaining awareness of its broader environment for safe navigation.

The mission objectives of the proposed system are to:

- Patrol and visually survey desk and nearby office areas to detect clutter or misplaced items
- Identify and classify at least ten common office objects using depth-based perception
- Decide and plan actions for each detected object based on its category and location
- Grasp and relocate desk-level items safely to designated zones or storage bins
- Navigate autonomously while maintaining situational awareness and avoiding obstacles
- Operate within defined safety limits, ensuring reliable motion control and human safety

These objectives collectively enable the development of a practical, intelligent robotic system that enhances workspace organization and safety in typical office environments.

Section 3: Mission and Task Decomposition

The autonomous office-desk organizing robot operates through a structured sequence of tasks that define how it interprets, plans, and executes actions within its workspace. Each task contributes to the overall mission of detecting and organizing common office items while maintaining safety, precision, and operational efficiency. The mission is carried out as a discrete organizing cycle, beginning with a start command and concluding when the desk and surrounding area are restored to an organized state.

The complete mission flow is illustrated as a finite-state machine in Figures 1 and 2, showing the eight operational states, key decision points, and loop-back conditions that enable adaptive behaviour and error recovery.

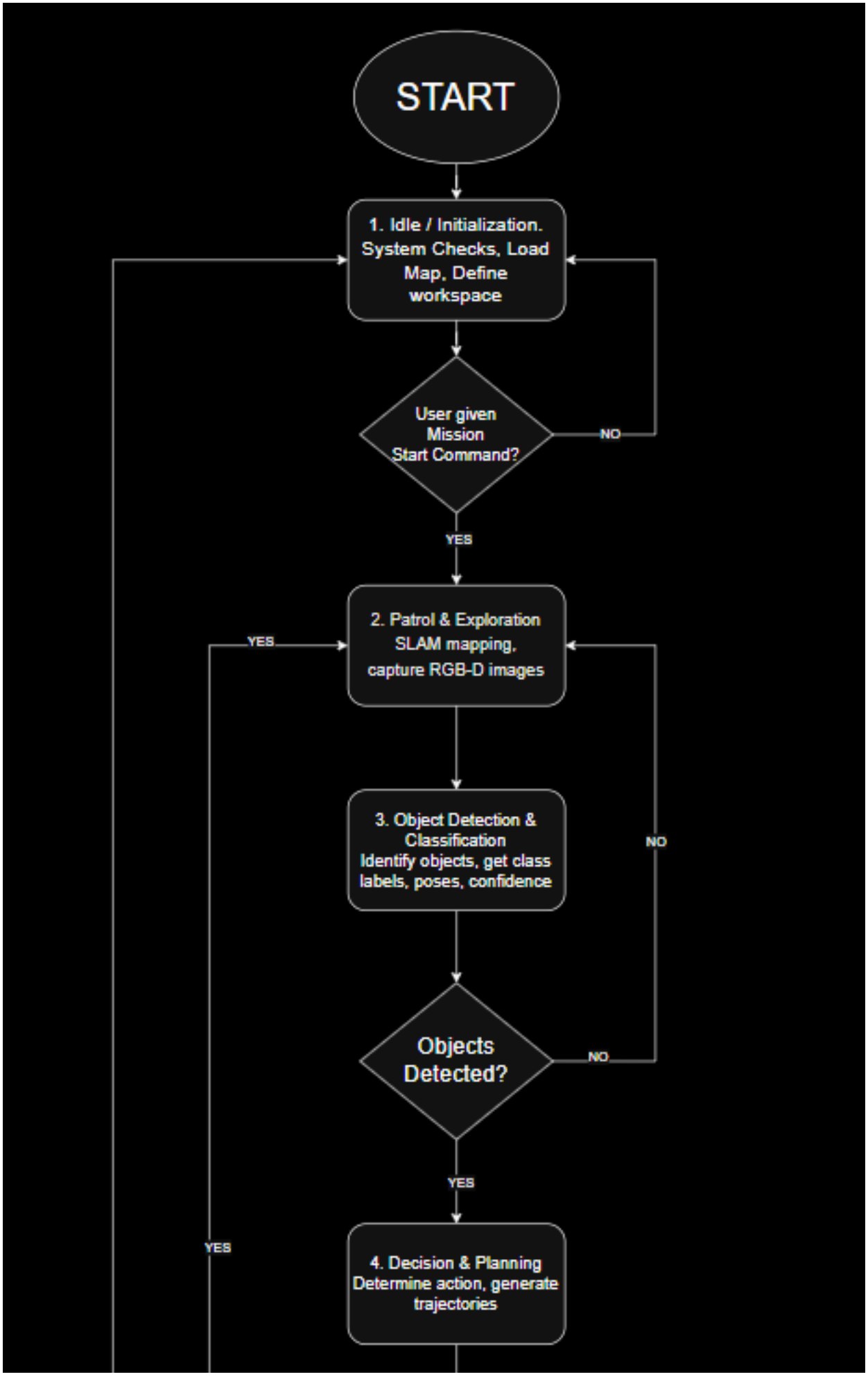

Figure 1 Mission flow state machine (Part 1) showing initialization, patrol, detection, and planning phases with decision points for mission start and object detection.

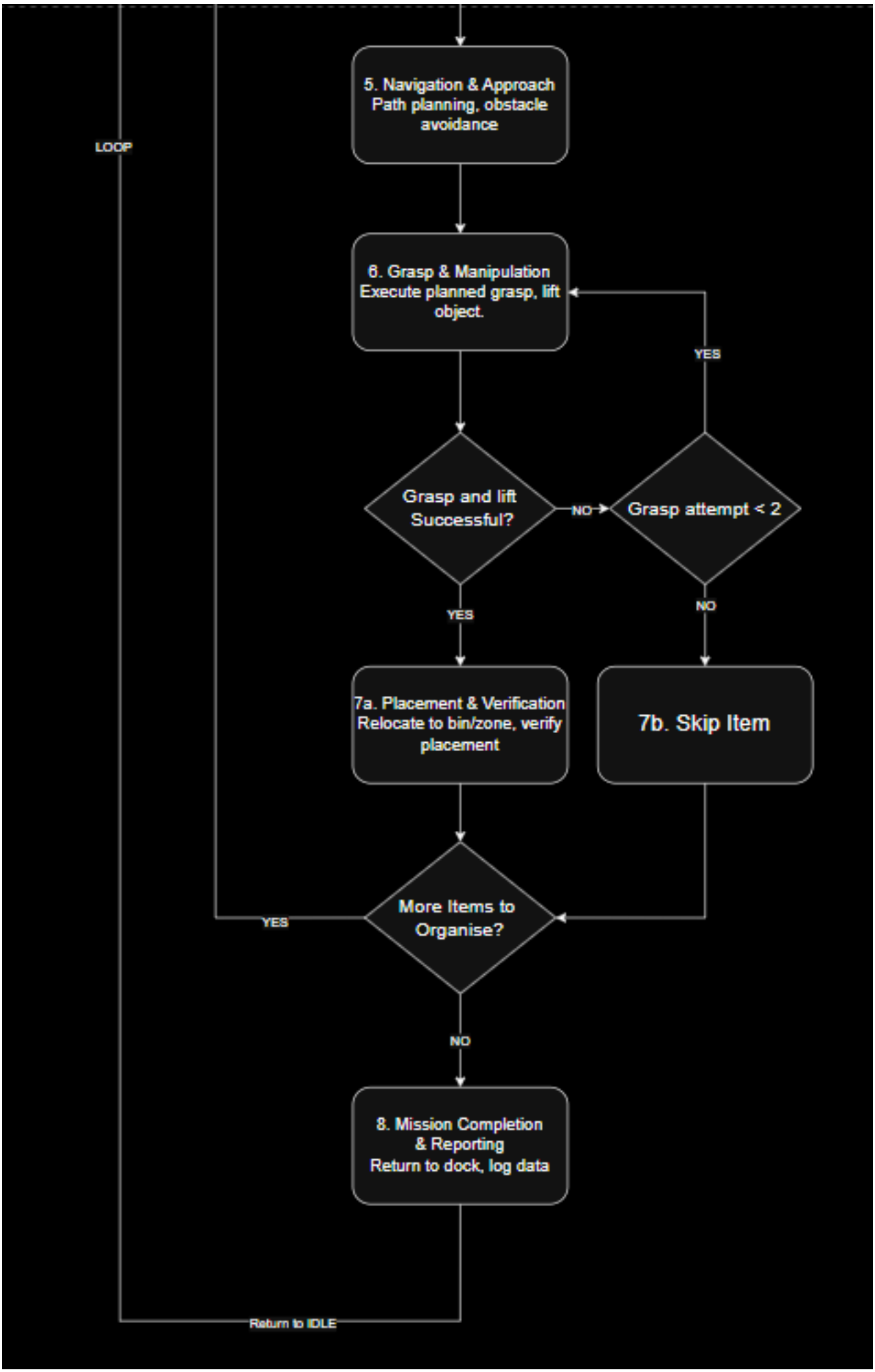

Figure 2 Mission flow state machine (Part 2) showing navigation, manipulation, placement, and completion phases with error recovery logic and loop-back conditions.

3.1 Mission Flow Overview

The mission can be represented as a finite-state process consisting of the following key stages:

1. Idle / Initialization:
The robot begins in a standby state, performing system checks, initializing sensors, and loading the environment map. Once the workspace boundaries and goals are defined, the mission starts upon receiving a start command from the operator.

2. Patrol and Exploration:
The robot patrols the designated desk area and nearby surroundings, using SLAM-based mapping to build or update its internal workspace map. It captures RGB-D images for analysis and identifies potential items of interest.

3. Object Detection and Classification:
Captured images are processed by the perception module to identify objects such as mugs, bottles, notebooks, or keyboards. Each detection produces class labels, confidence scores, and 3D pose estimates for grasp planning.

4. Decision and Planning:
Based on object type, position, and environmental context, the system determines whether to pick up, reposition, or leave an item. The task planner generates motion and grasp trajectories accordingly.

5. Navigation and Approach:
The mobile base navigates to the object's location while avoiding obstacles using path-planning and real-time sensor feedback.

6. Grasp and Manipulation:
The arm executes a planned grasp, verified by tactile or visual feedback, and lifts the object safely. If the grasp fails, the system attempts up to two retries before flagging the item for human intervention or skipping it entirely.

7. Placement and Verification:
The item is relocated to a predefined position or sorting bin. The system visually verifies that placement succeeded before proceeding.

8. Mission Completion and Reporting:
When all detected items are organized, the robot returns to its docking position and logs mission data for review. If additional items remain, the system loops back to the patrol phase to continue organizing.

Section 4. Required Skills and Interfaces

To achieve its mission objectives, the office-desk organizing robot integrates a range of coordinated skills—each implemented as a software and hardware module that communicates through defined interfaces. These modules collectively provide perception, planning, motion, manipulation, and safety capabilities essential for autonomous operation. Each skill is defined below with its respective inputs, outputs, and success criteria, along with the technical approach adopted to ensure reliable performance in real-world office environments.

4.1 Perception and Object Recognition

Function: Detect and classify multiple office items within the camera's field of view.

Inputs: RGB and depth images from RGB-D camera.

Outputs: Object class labels, bounding boxes, confidence scores, and 3D object poses.

Success Criteria: ≥85% detection accuracy (mAP@0.5) and ≤5 cm pose error.

Approach: The system employs YOLOv8-nano (YOLOv8n) for real-time object detection [10]. YOLO was selected over classification-only models (ResNet, MobileNet, EfficientNet) due to its ability to simultaneously detect and localize multiple objects in cluttered scenes—essential for office desks containing several items. YOLOv8n provides optimal performance on edge devices while maintaining accuracy. The model uses transfer learning on a custom-annotated dataset with bounding boxes for ten office item classes.

4.2 AprilTag Localization

Function: Detect AprilTags attached to fixed references (sorting bins, docking stations) for precise positioning.

Inputs: RGB image stream, known tag IDs and sizes.

Outputs: 6-DoF pose (position + orientation) of tags in camera frame; TF transforms published for navigation and manipulation.

Success Criteria: Tag detection within 0.5 s and <3 cm placement error.

Approach: Uses standard AprilTag detection libraries integrated with ROS 2 [9], [11]. Tags are placed only on stationary infrastructure elements to provide reliable spatial references for high-precision placement tasks, complementing the marker-less object recognition system.

4.3 Depth-to-3D and Pose Estimation

Function: Convert 2D bounding boxes from object detection to 3D coordinates for grasp planning.

Inputs: Bounding box coordinates, depth map, camera intrinsics.

Outputs: 3D centroid position and estimated grasp approach vector for each detected object.

Success Criteria: Pose accuracy sufficient for reliable grasp (≤10% mis-grasp rate).

Approach: Combines YOLO bounding boxes with aligned depth data to compute object centroids in 3D space. Camera calibration parameters transform pixel coordinates to world coordinates, enabling accurate reach planning for the manipulator arm.

4.4 Localization and Mapping (SLAM)

Function: Build and maintain a map of the workspace while localizing the robot within it.

Inputs: RGB-D or 2D LiDAR data, IMU readings, wheel odometry.

Outputs: Occupancy grid map and robot pose in map frame.

Success Criteria: Localization drift ≤0.5 m over 10 m travel; consistent obstacle representation.

Approach: Implements ROS 2-compatible SLAM algorithms (e.g., SLAM Toolbox or Cartographer) using sensor fusion [12], [13]. The map enables path planning and provides spatial context for identifying desk locations and navigation waypoints throughout the mission.

4.5 Navigation and Path Planning

Function: Plan and execute safe collision-free trajectories between waypoints.

Inputs: Occupancy grid map, current robot pose, goal pose.

Outputs: Velocity commands (cmd_vel) and planned path trajectories.

Success Criteria: Reach target without collisions; dynamic re-planning if obstacles appear.

Approach: Uses ROS 2 Navigation Stack (Nav2) with global and local planners [14]. Costmaps integrate static map data and real-time sensor input for dynamic obstacle avoidance. Maximum velocity is capped at 0.5 m/s for safety.

4.6 Manipulation and Grasp Control

Function: Control the 6-DOF robotic arm and gripper to pick, hold, and place items.

Inputs: Target 3D pose, grasp type, gripper state feedback.

Outputs: Joint trajectories, grasp confirmation signals.

Success Criteria: ≥80% grasp success rate and secure placement without slippage.

Approach: Employs MoveIt 2 motion planning framework for collision-free arm trajectories [15]. Grasp strategies are predefined per object class. Visual or tactile feedback verifies grasp success. Failed grasps trigger up to two retries before skipping the item.

4.7 Task Planning and Coordination

Function: Decide mission priorities and sequence tasks across all subsystems.

Inputs: Detected objects, current system state, mission goals, safety status.

Outputs: Ordered task list and action goals for navigation and manipulation.

Success Criteria: Efficient task completion with minimal idle time and safe operation throughout.

Approach: Implements a finite-state machine coordinating perception, navigation, and manipulation modules. The planner evaluates object priority based on type and position, issuing sequential commands via ROS 2 action servers.

4.8 Human–Robot Interaction (HRI) and User Interface

Function: Provide operators with controls, feedback, and system status visibility.

Inputs: Start/stop commands, mode selection, emergency stop trigger.

Outputs: Mission progress updates, alerts, error logs, visual feedback.

Success Criteria: Reliable manual override (<100 ms response) and clear system state communication.

Approach: Implements a graphical user interface (GUI) or command-line interface for mission control. Status messages and alerts are published via ROS 2 topics, enabling remote monitoring and intervention when necessary.

4.9 Safety and Diagnostics

Function: Monitor system health, enforce motion limits, and respond to emergency conditions.

Inputs: Sensor health data, proximity readings, battery voltage, actuator status.

Outputs: Safety flags, emergency stop commands, diagnostic logs.

Success Criteria: Immediate stop upon e-stop trigger (<100 ms response) and safe shutdown behaviour.

Approach: Continuous monitoring of all sensors and actuators with watchdog timers. Hardware e-stop button provides physical override. Software safety layer enforces velocity limits, minimum obstacle clearance (0.3 m), and battery thresholds to prevent unsafe operation.

Section 5. Hardware Design and Alternatives

The autonomous office-desk organizing robot is designed around a **Neobotix MP-500–based mobile manipulator platform**, providing the necessary payload capacity and mechanical stability to support the mounted UFactory xArm 6 and peripheral sensors. The MP-500's compact footprint and precise indoor navigation make it well suited for operation in controlled office environments. The hardware configuration prioritizes reliability, modularity, and ROS 2 compatibility, enabling seamless integration with perception, planning, and manipulation modules while maintaining a robust and scalable architecture.

5.1 Mobile Base

Primary: Neobotix MP-500 (ROS 2–compatible indoor differential-drive mobile base, ≈ 80 kg payload) [5].
Alternatives: Clearpath Husky (heavier outdoor platform, 75 kg payload) or TurtleBot 4 Pro (lower-cost educational model, 15 kg payload).
Justification: The Neobotix MP-500 provides an ideal balance of payload capacity, compact design, and precise indoor navigation. It safely supports the 12.2 kg UFactory xArm 6 and onboard hardware without overload. Unlike outdoor platforms such as the Clearpath Husky, the MP-500 is optimized for quiet, shared office environments with reduced noise and a smaller footprint. Its ROS 2-native integration and smooth differential-drive motion enable reliable SLAM, Nav2, and MoveIt 2 operation for safe, autonomous manipulation in confined workspaces.

5.2 Manipulator Arm

Primary: 6-DOF lightweight robotic arm (e.g., uFactory xArm) [1].
Alternatives: 4-DOF arm (simpler but limited reach and dexterity).
Justification: Six degrees of freedom enable versatile grasping, object orientation alignment, and collision-free motion planning via MoveIt 2.

5.3 End Effector

Primary: Parallel-jaw gripper with soft-finger pads.
Alternatives: Vacuum gripper (effective for smooth surfaces like phones); adaptive soft gripper (handles irregular shapes).
Justification: Parallel-jaw design offers mechanical simplicity and adequate grip precision for most office items, upgradable with compliant pads.

5.4 Perception Sensors

Primary: RGB-D camera (Intel RealSense D435 or OAK-D) for colour and depth perception; optional 2D LiDAR for enhanced mapping [2], [16].
Alternatives: Stereo camera (lower cost but reduced depth accuracy).
Justification: RGB-D provides reliable short-range depth perception essential for grasp planning and collision avoidance in desk environments.

5.5 Computation Platform

Primary: NVIDIA Jetson Orin Nano [4].
Alternatives: Intel NUC (higher compute, larger power draw); Raspberry Pi 5 with Edge TPU (budget option with offloaded inference).
Justification: Jetson Orin Nano balances GPU acceleration for YOLOv8n inference, power efficiency, and native ROS 2 compatibility.

5.6 AprilTags and Reference Markers

Small AprilTags (family 36h11) installed on sorting bins and docking stations enable <3 cm placement accuracy without altering the environment.

5.7 Power and Safety Hardware

Primary: Rechargeable lithium-ion battery (2–3 hour operation); hardware emergency-stop button.
Alternatives: Swappable battery modules (extend mission endurance).
Justification: Integrated battery provides sufficient runtime for typical missions while e-stop ensures immediate user override.

Section 6. Software Architecture

The software architecture follows a modular, layered design implemented using ROS 2 [17]. Each functional skill operates as an independent node communicating through standardized topics, services, and actions, ensuring modularity and scalability.

6.1 System Layers

The software is divided into five core layers, as illustrated in the architecture overview (Figure 3) with detailed component specifications shown in Figures 4-8:

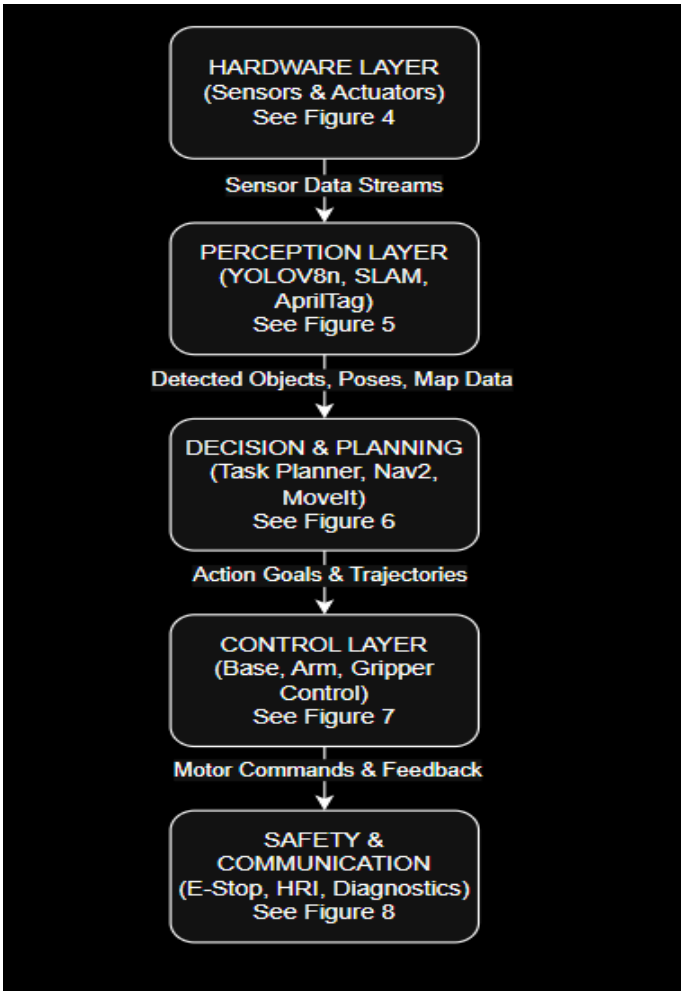

Figure 3 Software architecture block diagram showing the five-layer ROS 2 system design with hardware sensors and actuators, perception modules, decision and planning components, control layer, and safety communication layer.

Hardware Layer (Figure 4): Physical sensors and actuators including RGB-D camera, optional 2D LiDAR, IMU, mobile base, 6-DOF manipulator arm, and gripper.

Perception Layer (Figure 5): Handles image acquisition, YOLOv8n object detection, AprilTag recognition, depth processing, and SLAM-based mapping and localization.

Decision and Planning Layer (Figure 6): Interprets perception data, prioritizes tasks, and coordinates motion planning for navigation (Nav2) and manipulation (MoveIt 2).

Control Layer (Figure 7): Executes real-time actuation for the mobile base, arm, and gripper using sensor feedback and collision avoidance. Safety and

Communication Layer (Figure 8): Manages user input, safety overrides, system diagnostics, and mission reporting through the HRI interface.

This modular hardware–software layering supports future scalability, enabling substitution of components such as different robotic arms or upgraded compute modules without

Figure 4 Hardware Layer - sensors and actuators for perception and manipulation.

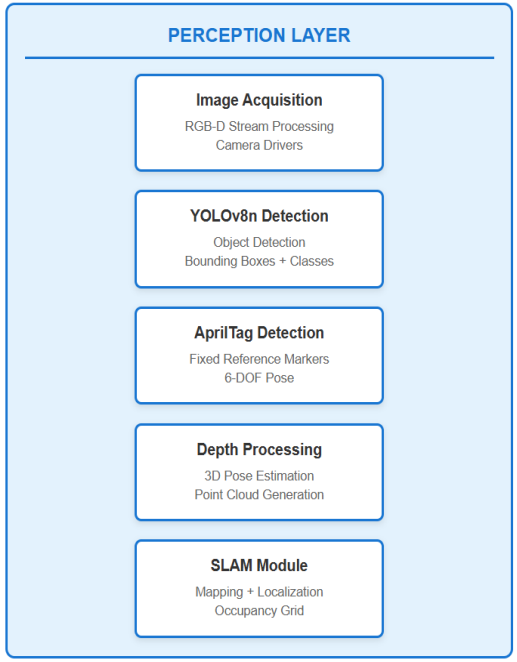

Figure 5 Perception Layer - image processing, object detection, and localization modules.

Figure 6 Decision and Planning Layer - task coordination and motion planning.

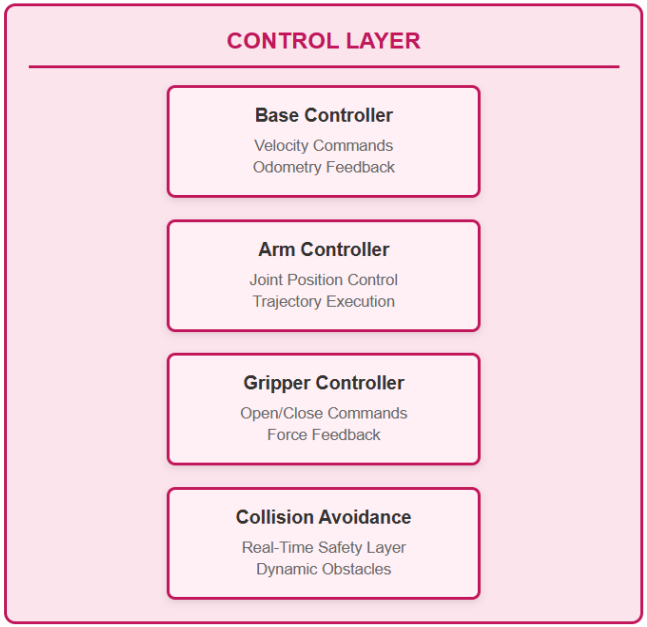

Figure 7 Control Layer - real-time execution and collision avoidance.

Figure 8 Safety and Communication Layer - monitoring, diagnostics, and user interface.

6.2 Key ROS 2 Interfaces

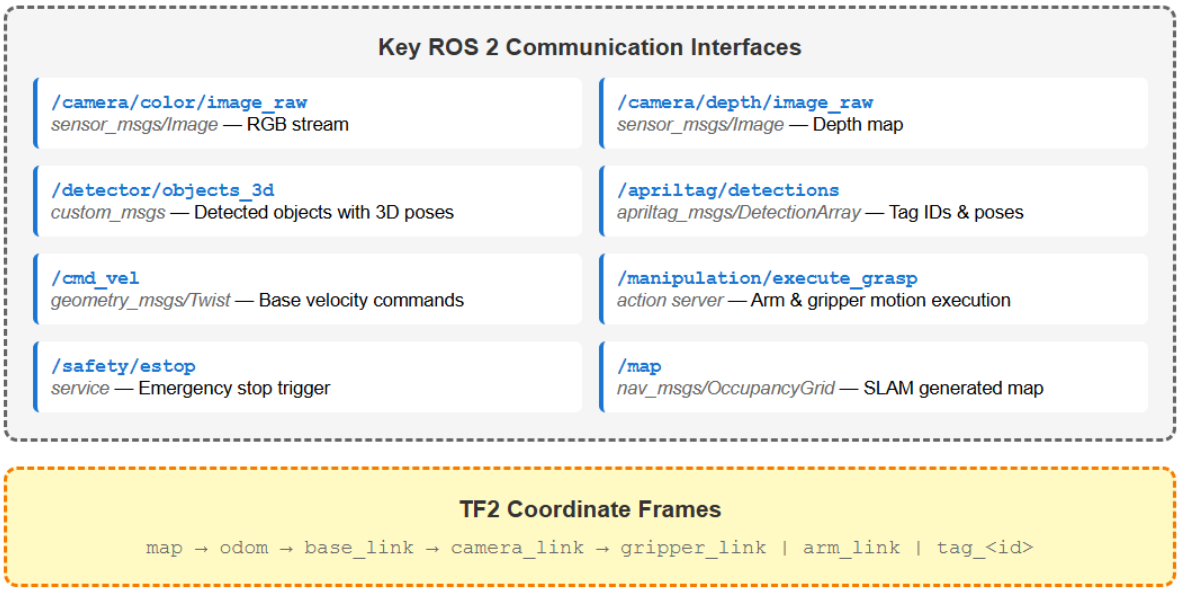

Figure 9 Key ROS 2 communication interfaces showing primary topics with message types for sensor data, detection results, control commands, and safety triggers, along with TF2 coordinate frame hierarchy.

The layers communicate through standardized ROS 2 interfaces as illustrated in Figure 9. These include sensor data streams (camera images, depth maps), perception outputs (detected objects with 3D poses, AprilTag detections, SLAM maps), control commands (base velocity, manipulation actions), and safety services (emergency stop). Coordinate frames (map, odom, base_link, camera_link, gripper_link) are managed via the tf2 transform system to maintain spatial relationships throughout the system.

6.3 Task Coordination

The Task Planner Node implements a finite-state machine (Figures 1, 2) coordinating all modules as shown in the architecture overview (Figure 3)

Section 7. Dataset and Model Plan

The perception system recognizes **ten office item classes**: mug, water bottle, mobile phone, keyboard, mouse, stapler, pen/pencil, notebook, office chair, and bin. The first eight are primary organizing targets; the last two provide environmental context for navigation.

Dataset Collection: At least 2000 samples per class sourced from custom RGB-D captures under varied lighting and clutter conditions, supplemented by filtered public datasets (COCO, OpenImages). Images are annotated using Roboflow or LabelImg in YOLO format with bounding boxes and class labels [18], [19]. The dataset follows an 80/10/10 train/validation/test split.

Model Architecture: YOLOv8n with ImageNet-pretrained backbone, fine-tuned via transfer learning at 416×416 resolution. Deployment options include native PyTorch (.pt), ONNX for portability, or TensorRT for optimized real-time inference on Jetson Orin Nano.

Evaluation Metrics: Performance assessed using accuracy, macro-F1 score, precision, recall, confusion matrix, and mAP@0.5 for detection quality. Target: ≥85% overall accuracy,

≥80% macro-F1, ≥75% per-class recall. Error analysis guides dataset rebalancing and model refinement.

Section 8. Risk and Safety Assessment

Safety is critical for autonomous systems in human-populated environments. The robot integrates multiple hardware and software safety layers to ensure reliable operation.

Operational Safety: Linear velocity capped at 0.5 m/s with safe arm joint speeds [20]. Hardware emergency-stop button mounted on chassis provides immediate actuator shutdown (<100 ms response). Virtual e-stop available via user interface. Minimum obstacle clearance of 0.3 m maintained during navigation through LiDAR and depth-based collision avoidance with dynamic re-planning.

Manipulation Safety: Grasp success verified via visual or tactile feedback before lifting. Failed grasps trigger up to two retries before flagging human intervention. Depth sensing maintains obstacle awareness within arm reach volume.

System Reliability: All sensors monitored through ROS 2 diagnostic nodes. Sensor failures trigger safe idle state. Battery voltage monitoring initiates low-power return-to-dock sequence. Human presence detection slows or pauses operation in shared spaces, ensuring compliance with privacy and safety standards.

Section 9. Budget and Bill of Materials

The proposed system is designed to balance performance and cost-effectiveness using commercially available components. The bill of materials (Table 1) reflects current UK market prices and prioritizes modularity for ease of replacement and upgrades.

Table 1 Bill of Materials (BoM)

Component	Specification	Unit Price (GBP)	Qty	Total (GBP)	Source
6-DOF Robotic Arm	uFactory xArm 6 (5kg payload, 700mm reach)	£7,300	1	£7,300	[1]
Parallel-Jaw Gripper	Bio Gripper G2	£ 1,370	1	£ 1,370	[1]
RGB-D Camera	Intel RealSense D435	£250	1	£250	[2]
2D LiDAR (Optional)	RPLIDAR A1M8 (12m range, 360°)	£80	1	£80	[3]
Computation Platform	NVIDIA Jetson Orin Nano Super (67 TOPS)	£220	1	£220	[4]
Mobile Base	Neobotix MP-500 (ROS 2 compatible, ≈ 80 kg payload)	~£9,000 (est.)*	1	~£9,000	[5]
IMU	BMI055 or equivalent	£20	1	£20	[6]
Battery Pack	Li-ion rechargeable (2-3 hour runtime)	£75	1	£75	[7]
E-Stop Button	Hardware emergency stop	£15	1	£15	[8]
AprilTags	Printed tags (family 36h11)	£0	10	£0	[9]
Miscellaneous	Cables, mounting hardware, connectors	£150	-	£150	-
			Total Estimated Cost:	**~£19,130**	

N/B: Estimated cost based on vendor quotations (Neobotix GmbH, 2025). Actual pricing available upon request.

Section 10. Conclusion

This report presents the design of an autonomous office organization robot integrating AI-driven perception, manipulation, and navigation capabilities. The system employs YOLOv8n for real-time multi-object detection, ROS 2 for modular software architecture, and MoveIt 2 for safe manipulation planning. Hardware selection prioritizes industrial-grade reliability through the Neobotix MP-500 mobile base and UFactory xArm 6, while maintaining cost-effectiveness with the NVIDIA Jetson Orin Nano for edge AI inference.

The proposed design targets ≥85% object detection accuracy and ≥80% grasp success rate, with comprehensive safety mechanisms including emergency stop controls, velocity limits, and collision avoidance. The modular architecture enables scalability and component substitution, making the system viable for research, education, and prototype development. Future work includes real-world validation, adaptive grasp learning, multi-robot coordination, and extended object classification to enhance workspace automation capabilities in dynamic office environments.

Section 11. References:

[1] UFactory xArm 6: Robotic Arm

[2] Intel RealSense D435: RBGD - Camera

[3] RPLIDAR A1M8: LIDAR

[4] NVIDIA Jetson Orin Nano: Jetson Orin

[5] Mobile Base: NeoBotix Mobile Base

[6] IMU BMI055: BMI055 Inertial Measurement Units - Bosch | DigiKey

[7] Battery Pack: Battery Pack (Li-ion battery packs)

[8] E-Stop Button: Emergency Stop (emergency stop buttons)

[9] AprilTags: https://github.com/AprilRobotics/apriltag (free printable)

[10] Ultralytics. "YOLOv8: State-of-the-Art Real-Time Object Detection." *Ultralytics GitHub Repository*, 2024. Available: YOLO

[11] Olson, E. "AprilTag: A robust and flexible visual fiducial system." *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[12] Nav2 Documentation. Available: Navigating while Mapping (SLAM) — Nav2 1.0.0 documentation

[13] Google Cartographer. "Google Cartographer ROS Integration." Available: https://google-cartographer.readthedocs.io/en/latest/

[14] Nav2 Documentation. Available: Nav2 — Nav2 1.0.0 documentation

[15] MoveIt Project. "MoveIt 2: Motion Planning Framework for ROS 2." *MoveIt Documentation*, 2024. Available: https://moveit.ros.org/

[16] Luxonis. "OAK-D DepthAI Platform." *Luxonis Documentation*, 2024. Available: https://docs.luxonis.com/

[17] Open Robotics. "Robot Operating System 2 (ROS 2) Documentation." *ROS.org*, 2024. Available: https://docs.ros.org/

[18] Roboflow. "Roboflow: The Computer Vision Platform." 2024. Available: https://roboflow.com/

[19] Tzutalin. "LabelImg: Image Annotation Tool." *GitHub Repository*, 2015. Available: https://github.com/tzutalin/labelImg

[20] International Organization for Standardization. "ISO 10218-1:2011 — Robots and robotic devices — Safety requirements for industrial robots." ISO, Geneva.