Introduction and Objectives

- **Project Overview:**
  We are developing a multi-robot delivery system designed to operate indoors in a simulated office environment.

- **Assignment 3 Focus:**

  - Refine and prioritize system requirements based on client feedback.

  - Identify missing software components and choose suitable tools and frameworks.

  - Implement a prototype with key functionalities for robot deployment, docking, and task interaction.

Core Functionalities

- **Robot Deployment:**
  Spawn two or more TurtleBot4 robots at different locations in the environment, typically docking stations.

- **Docking Stations:**
  Multiple docking stations configured; robots can dock to any station, not one-to-one mapped.

- **Interaction Logic:**
  Interface for specifying delivery tasks with pickup and drop-off locations.
  Functionality to confirm task execution through user acknowledgments on pickup and delivery.

System Components Overview

- **Robot Navigation:**
  ROS 2 Nav2 stack providing path planning, obstacle avoidance, and localization.

- **Simulation Environment:**
  Gazebo world created using FloorPlan-DSL models representing the office layout.

- **Task Interaction:**
  Simple command-line interface to assign tasks and confirm completions.

- **Docking Management:**
  Docking stations defined as named locations in configuration files.

- **Missing Components:**
  Custom logic for task management and robot-to-task assignment developed by the team.

Tools and Libraries Used

- **ROS 2:** Middleware facilitating communication and control of robots.
- **Nav2 Stack:** Navigation and localization algorithms for autonomous movement.
- **Gazebo:** 3D robot and environment simulation.
- **FloorPlan-DSL:** Domain-specific language used to model and generate the office layout.
- **Docker & textX:** For model generation and environment setup automation.
- **Teleop_twist_keyboard:** Manual teleoperation tool used during testing.
- **RViz2:** Visualization tool for robot pose, sensor data, and environment mapping.

Software Architecture Diagram

Diagram Description:

- Environment modeled by FloorPlan-DSL → Gazebo simulation.
- Robots run ROS 2 nodes integrating Nav2 navigation, localization, and interaction logic.
- Task interface module handles delivery task assignment and confirmations.
- Docking stations configured in a shared map accessible by all robots.
- Visualization through RViz2, showing robot poses and sensor data.

Prototype Implementation Details

- **Robot Deployment:**
  Robots are spawned at docking stations defined in the configuration file at simulation start.

- **Docking Stations:**
  Configured as named locations in a YAML file allowing flexibility; robots dock to any available station.

- **Task Interface:**
  Command-line interface lets users create delivery tasks by specifying pickup and drop-off locations.

- **Confirmation Flow:**
  After task execution, users confirm pickups and deliveries to simulate task acknowledgment.

- **Multi-Robot Coordination:**
  Basic assignment ensures tasks are allocated to the appropriate robot based on location and availability.