**Roadmap for "RobotGPT" Project**

**Objective:**

To create an autonomous robotic system capable of detecting, selecting, and grasping a user-specified object from a workspace. The user inputs the object name, and the system handles the task from object recognition to grasping it.

**1. Detection Team: Object Recognition and Workspace Mapping**

* **Goal**: Develop a robust system to detect objects in the workspace based on the user's input.

**Tasks:**

* + **Object Identification**:
    - Develop a database or library of common objects with labels that can be referenced.
    - Train or integrate object recognition models (e.g., using deep learning with YOLO, Faster R-CNN, or similar algorithms).
    - Implement a camera system for 3D object detection and workspace monitoring (e.g., using depth cameras, stereo vision, or LiDAR).
  + **Object Localization**:
    - Ensure the object detection system provides precise object location (coordinates) in 3D space.
    - Account for occlusions or multiple objects in the field of view.
    - Optimize the detection speed to allow real-time performance.
  + **Workspace Mapping**:
    - Use SLAM (Simultaneous Localization and Mapping) or a similar approach to dynamically update the workspace.
    - Ensure the robot understands its environment to avoid collisions with non-target objects.

**2. Manipulation Team: Path Planning and Motion Control**

* **Goal**: Create a motion planning system that moves the robotic arm from its current position to the target object without collision.

**Tasks:**

* + **Motion Planning Algorithms**:
    - Implement path planning algorithms such as RRT (Rapidly-exploring Random Tree) or PRM (Probabilistic Roadmap) for obstacle avoidance.
    - Optimize for the shortest path and safest movement.
  + **Coordinate Transformation**:
    - Translate object coordinates from the detection team to the robot's coordinate system.
    - Use inverse kinematics (IK) to calculate joint movements of the manipulator to reach the object.
  + **Collision Avoidance**:
    - Integrate sensor feedback to avoid obstacles dynamically during motion.
    - Create a feedback loop for real-time path adjustments.
  + **Simulation**:
    - Before deployment, run tests in a simulated environment (e.g., using Gazebo or V-REP) to verify the motion planning.

**3. Grabbing Team: Object Grasping and Feedback Control**

* **Goal**: Enable the robotic hand (end-effector) to grip the object securely after the manipulator reaches it.

**Tasks:**

* + **Gripper Control**:
    - Develop control algorithms for the gripper (robotic hand), ensuring it adapts to different shapes and sizes of objects.
    - Implement force sensing or tactile feedback to ensure the object is securely gripped without being damaged.
  + **Grasp Planning**:
    - Use geometric and physical properties of objects (e.g., size, shape, weight) to determine the best grasping point.
    - Explore grasping algorithms such as the GPD (Grasp Pose Detection) method or DexNet.
  + **Pick and Place Logic**:
    - Once an object is grasped, create a mechanism to confirm the success of the operation (feedback control).
    - Develop logic for placing the object at a user-defined location or storing it in a safe area.
  + **Testing and Calibration**:
    - Ensure that the gripper is calibrated for various object sizes and materials.

**Overall System Integration**

* **User Input Interface**:
  + Create a user-friendly interface where users can input the name of the object to be picked (e.g., a simple web-based interface or text input console).
  + The system should translate the user's command into corresponding actions (triggering detection, manipulation, and grabbing).
* **Robust Error Handling**:
  + Develop mechanisms for handling detection failures (e.g., object not found), grasping failures, and other issues.
  + Use fallback strategies (e.g., retry or request manual intervention if necessary).

**Milestones and Timeline:**

1. **Week 1-2**: Object detection and workspace mapping setup.
2. **Week 3-4**: Motion planning algorithms and inverse kinematics development.
3. **Week 5-6**: Gripper control and grasp planning implementation.
4. **Week 7**: Full system integration and testing in a simulated environment.
5. **Week 8**: Live testing and final system optimization.