



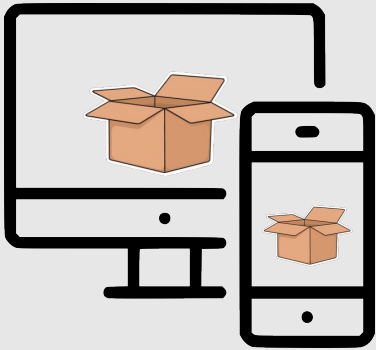
Retriever Rover

Design Review

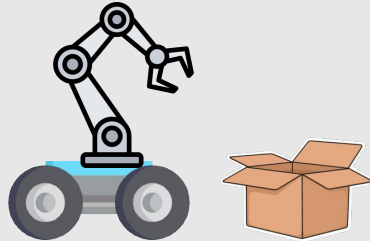
Team C0: **Jasmine Chao**, Matthew Mao, Joonyoung Huh

Use Case/Application

Nearly a quarter of adults have trouble bending down or kneeling, according to PMC

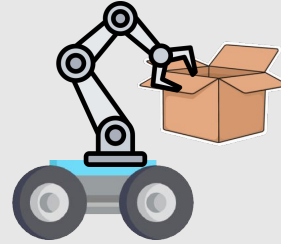


1. User will send a request via WebApp to retrieve an object (boxes/grocery bags)



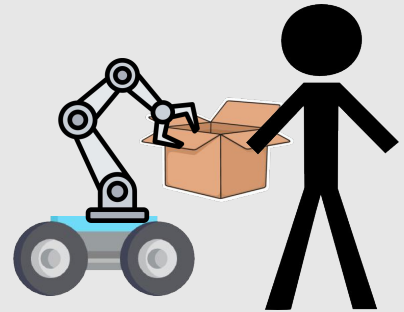
2. Robot will navigate and locate object

95% Accuracy



3. Robot will pick up object

90% Accuracy



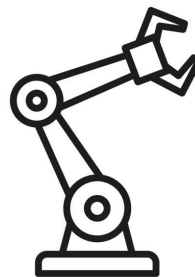
4. Robot will find user and return object at an acceptable height

85% Accuracy

Changes

What we changed since Proposal Presentation:

- Ramp
 - Not practical
 - Change: Have robot find user and have a robotic arm that can lift object up to user
- Jetson for motor control
 - Real-time, safety, modularity
 - Change: Use Arduino Nano ESP32 for motor control
- Picking up pill bottles, wallets, phones, etc.
 - Too much variance
 - Change: Pick up boxes/grocery bags



Quantitative Design Requirements

Object Detection

- Detects object in $\leq \underline{1s}$
- Detects object from $\leq \underline{0.5m}$

Lifting & Delivery

- Payload capacity $\geq \underline{150 \text{ grams}}$
- Delivery height $\geq \underline{0.5m (20 \text{ inches})}$

Rover

- Speed $\geq \underline{1m/s}$
- Turning radius = $\underline{0m}$ (In site rotation)
- Runtime $\geq \underline{10 \text{ minutes}}$

User Interface

- Command latency $\leq \underline{3s}$
- Feedback refresh rate $\geq \underline{5Hz}$

Solution Approach

Public Health, Safety, Welfare

- Reduces fall risk from bending and carrying heavy objects
- Autonomous obstacle avoidance prevents collisions
- Prioritize stable mechanical design
- Stopping is the safe state of the robot

Economic Factors

- \$600 budget
- Commercially available components
- Modular design to switch out components for future improvements

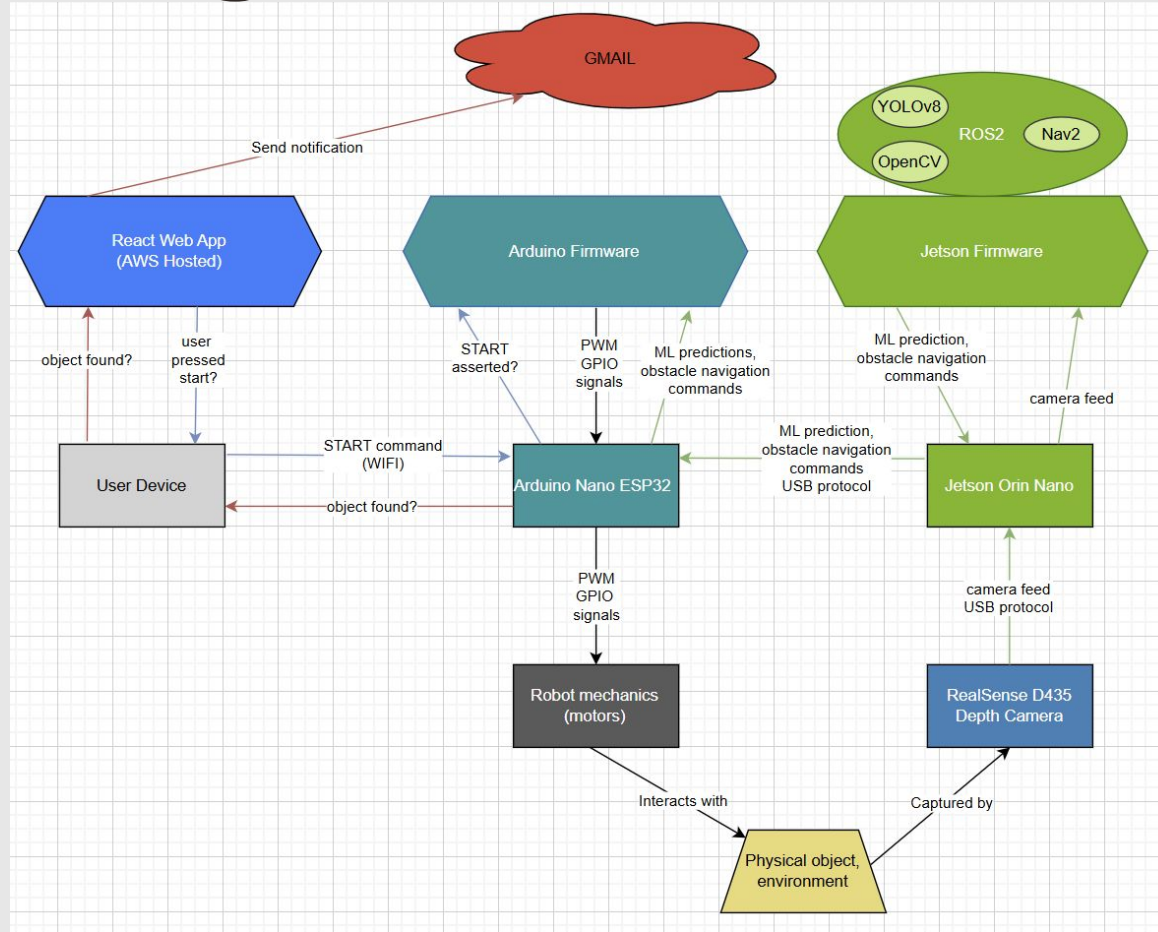
Social/Cultural Factors

- Preserves independence of users
- Designed to assist caretakers, not take away jobs

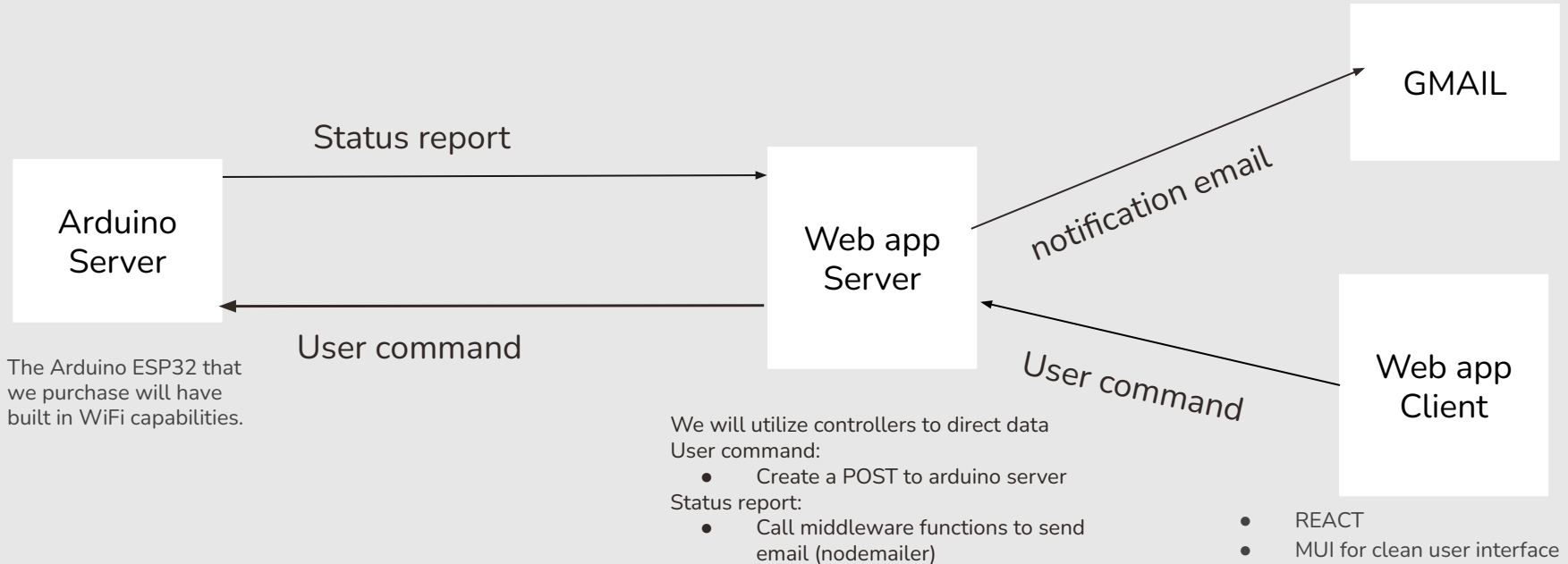
Environmental Factors

- Low energy consumption (WiFi)
- Reusable, modular components
- Compact, indoor robot

Block Diagram

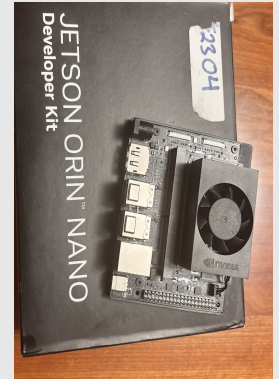
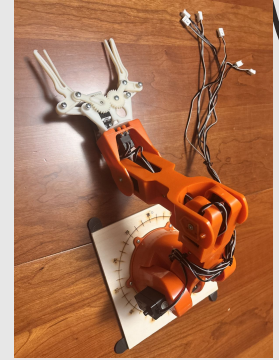


Implementation Plan - Server/Web App



Implementation Plan - Rover and Arm

- Currently, we have a robot arm and drivetrain from inventory
- We will remove the wooden base from the robot arm and attach it to the front of the robot base using the through holes
- We will attach pieces of rubber to the claw to increase friction
- In the back of the drivetrain, we will mount our Jetson Orin Nano and Arduino Nano ESP32.
- We will place our camera on the base for consistency.



Testing Verification and Metrics -Robot

Requirements	Testing	Expectation Metrics
Object detection	Place target object at random floor locations and run detection over 100 trials across 3 distances (0.5m, 1m, 2m) under 3 lighting conditions	$\geq 95\%$ detection accuracy
Object Grasping	Perform 100 repeated pickup trials for each object type, varying object orientation $\pm 30^\circ$	$\geq 90\%$ successful grasps without object damage
Navigation to the user(delivery)	Allow the robot to pick up an object and attempt to travel to and to the user. Perform 100 trials, and each trial starts when the robot has successfully picked up an object	$\geq 85\%$ successful navigation without dropping the object, or flipping over

Testing Verification and Metrics - Web App

Requirements	Testing	Expectation Metrics
User Notification	During our trials for detection and delivery, we will also keep track of whether or not notifications are being sent once robot picks item up	100% successful notification delivery
Low Latency (App ↔ Robot)	For 20 trials, we will measure the time from user submitting target via web app to robot initiating motion, and from pick up to user receiving notification	Command latency ≤ 3 s Notification latency ≤ 3 s

Risk Mitigation

Risks	Mitigations
Object detection failure	• Adjust confidence threshold and require 3-frame detection consistency. • If object is not detected within 5 seconds, perform 360° scan
Robot getting stuck	• If monitored velocity < 0.1 m/s for > 3 seconds, initiate recovery maneuver: reverse 0.3 m and replan path • After two failed attempts, abort mission and notify user
Gripper slip	• Use rubberized gripping surfaces • Maintain grip force $\geq 1.5\times$ required holding force

Project Management

Retriever Rover

Team C0

Project start: **Fri, 1/30/2026**

Display week: **1**

TASK	ASSIGNED TO	PROGRESS	START	END
Logistics				
Website Setup	All	100%	1/19/26	1/31/26
Proposal Present	All	100%	2/1/26	2/6/26
Design Presentation	All	0%	2/15/26	2/20/26
Interim Demo	All	0%	3/30/26	4/3/26
Final	All	0%	4/27/26	5/1/26
Project Planning				
Order Materials	All	0%	2/20/26	2/27/26
Finalize Design	All	0%	2/8/26	2/14/26
Project				
Train YOLOv8 or Joon		0%	3/9/26	3/12/26
Train YOLOv8 or Joon		0%	3/12/26	3/15/26
Train YOLOv8 or Joon		0%	3/15/26	3/18/26
Slack	Joon	0%	3/18/26	3/23/26
Assemble Robot	Matthew	0%	3/9/26	3/13/26
Make grippers w/ Matthew		0%	3/13/26	3/20/26
Slack	Matthew	0%	3/20/26	3/23/26
Setup ROS2	Jasmine	0%	3/9/26	3/16/26
Setup Nav2	Jasmine	0%	3/13/26	3/20/26
Slack	Jasmine	0%	3/20/26	3/23/26
OpenCV with YO Jasmine and Joon		0%	3/23/26	3/30/26
Test and refine c	Matthew	0%	3/23/26	3/30/26
Slack	All	0%	3/30/26	4/3/26
Make app	Joon	0%	4/3/26	4/13/26
Make notification	Matthew	0%	4/3/26	4/13/26
Combine app + r	Matthew and Joon	0%	4/13/26	4/17/26
Construct Ramp	Jasmine	0%	4/3/26	4/10/26
Slack	Jasmine	0%	4/10/26	4/17/26
Testing/Debugging	All	0%	4/17/26	4/24/26
Slack	All	0%	4/24/26	4/26/26

