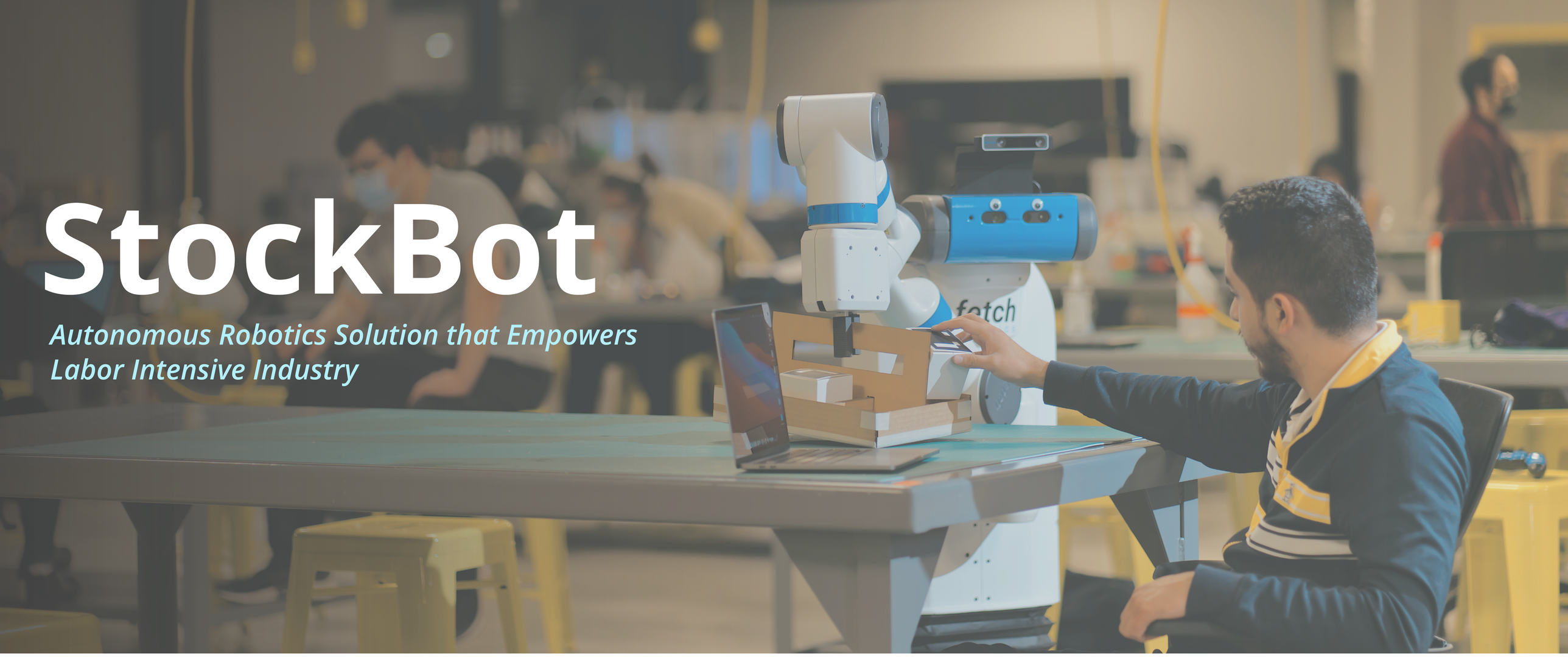


StockBot

Autonomous Robotics Solution that Empowers Labor Intensive Industry

Problem

Equipment management consumes a large amount of labor at many companies, including warehouses, factories, laboratories, hospitals, and libraries. Employees spend a lot of time looking for, fetching, transporting, delivering, retrieving, managing, and organizing equipment (e.g. products, tools, AV equipment, medical instruments, books, etc.). That time could be better spent on other, more intellectually challenging aspects of their jobs.

Another problem with conventional equipment management and delivery system is that it is not efficient for an employee to streamline the collecting, kitting and delivering pipeline.

Solution

Our proposed solution to this problem is a fully automated equipment management and delivery system. Our solution includes automated storage cells, autonomous delivery robots, a web interface for requesting items, and a backend to connect and manage the other components.

The delivery robot is a Fetch Mobile Manipulator, a 7-DOF robotic arm sitting on a moving base. This robot can pick up a tray full of items, navigate from the storage system to the user (and vice versa), and place the tray at the intended location. The navigation functionality is accomplished using an Intel RealSense T265 tracking camera, 2D LIDAR scanner, and ArUco fiducial markers, as well as the open-source ROS (Robot Operating System) navigation stack. The grasping functions utilize a 3D Depth Camera and the ROS MoveIt motion planning library.

The storage system consists of a Kinova Gen3 robotic arm that can pick up boxes holding items from a storage space and place them into a tray (and vice versa). The boxes, as well as the tray and storage spaces, are marked with ArUco fiducial markers. The grasping program utilizes OpenCV for ArUco marker detection and the Kinova Kortex library for motion planning.

The web interface enables users to request and return items. It is accessible from any device with a web browser. The front-end site was built using HTML and CSS. The back-end was programmed in Python utilizing the Flask web framework.

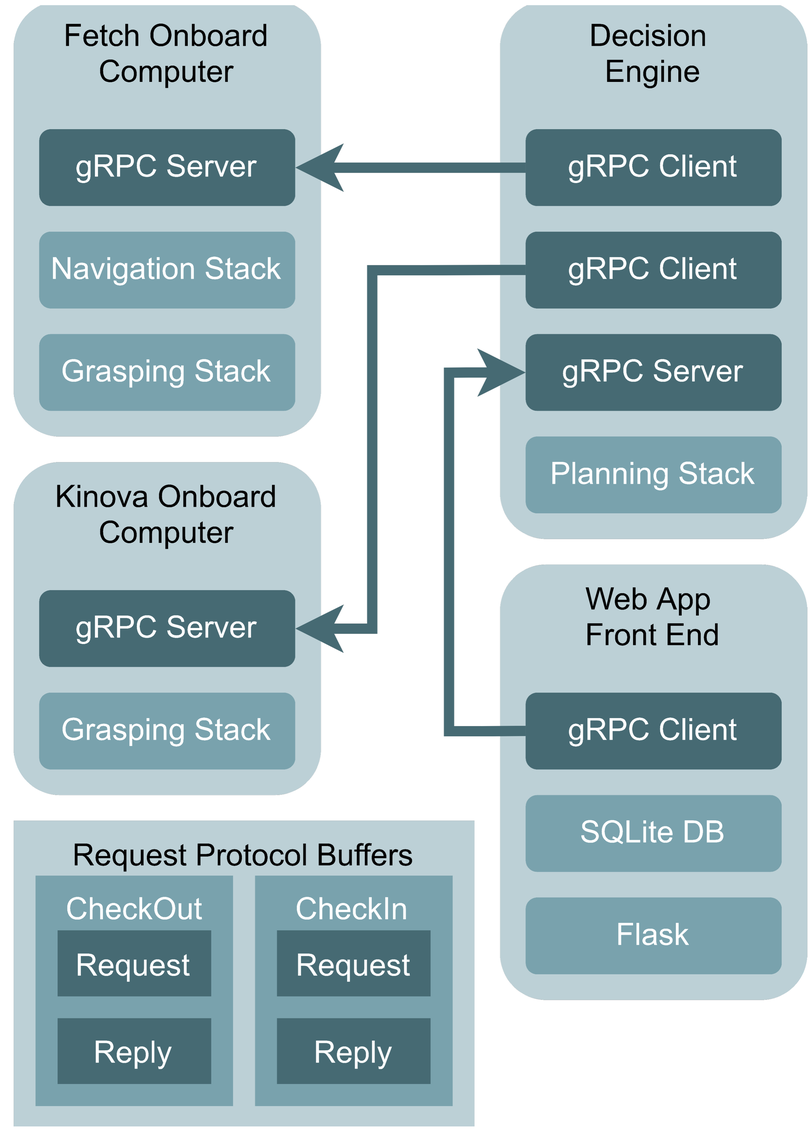
Our decision engine is the main server that communicates with the other three components (delivery robot, storage robot, and web application). It was built using Python and uses the gRPC protocol to communicate with the other components.

Design Process

We conducted primary and secondary research to understand which features and information users need during the check-in and check-out process. After building our first prototype, we conducted several usability tests to determine how end users interacted with the mobile delivery robot and the web interface.

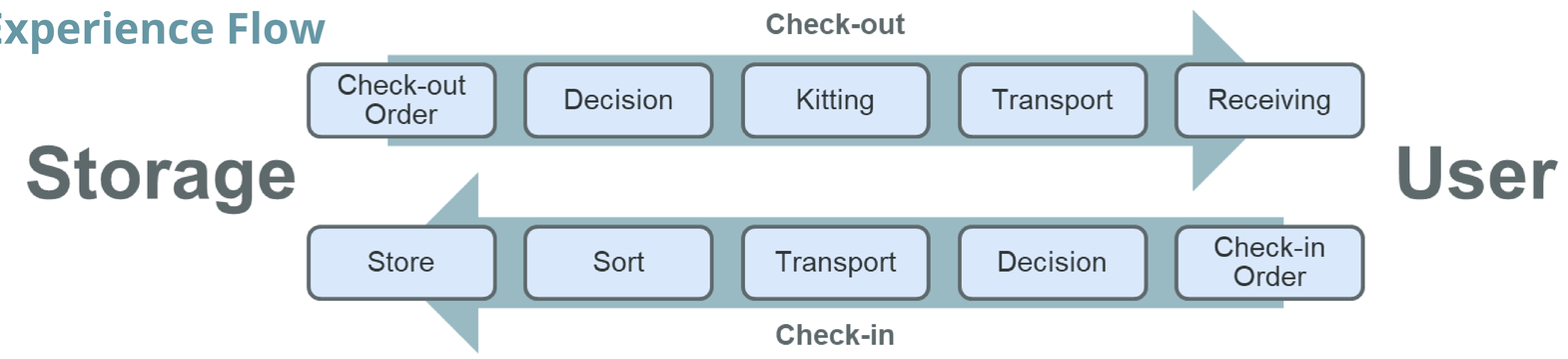
We also evaluated the functionality of our robotic system to ensure that it met our design requirements. Our final design incorporated the lessons that we learned from our repeated design-build-test iterations.

Kinova and Fetch Collaboration



System Architecture

Experience Flow



GLOBAL INNOVATION EXCHANGE

fetch robotics

KINOVA

C4RT

Annan Li
Daniel Azúa
James Muir
Sulaiman Al Akbari
Yifei Fang