Department of Computer Science & Engineering
The University of Texas at Arlington

System Requirements Specification
CSE 4316: Senior Design I
Spring 2025

THE SUCTIONEERS
THE VACU-ARM

Team Name
Product Name

Andrew Weiler
Andrew Whitmill
Jonathan Camarce
Samriddha Kharel
Seezayn Bishwokarma

Revision History

Revision	Date	Author(s)	Description
0.1	8.01.2025	AW, AW, JC, SK	first draft
1.0	12.04.2025	AW	final draft

Contents

List of Figures

1 Product Concept

For our senior design project, we are building the Vacu-Arm, a robotic arm that uses suction to pick up and move objects. It will automate the process of transferring items quickly and accurately. This device is designed to reduce the need for manual handling. By doing so, it will help minimize human error and physical labor.

1.1 Purpose and Use

The Vacu-Arm will pick up objects using suction and automatically move them to a new location. It is designed to handle small to medium items safely and quickly. Users can set it up to perform repetitive tasks such as sorting, packing, or transferring parts. It should reduce manual lifting and improve accuracy in simple handling tasks.

1.2 Intended Audience

The Vacu-Arm is designed for workshops, and light industrial settings that need to automate simple picking and placing tasks. It could be used by manufacturers, warehouse operators, or anyone handling repetitive sorting or packaging work. The product is designed for users who want a tool to reduce manual labor. It can work as a standalone unit or as part of a larger automated system.

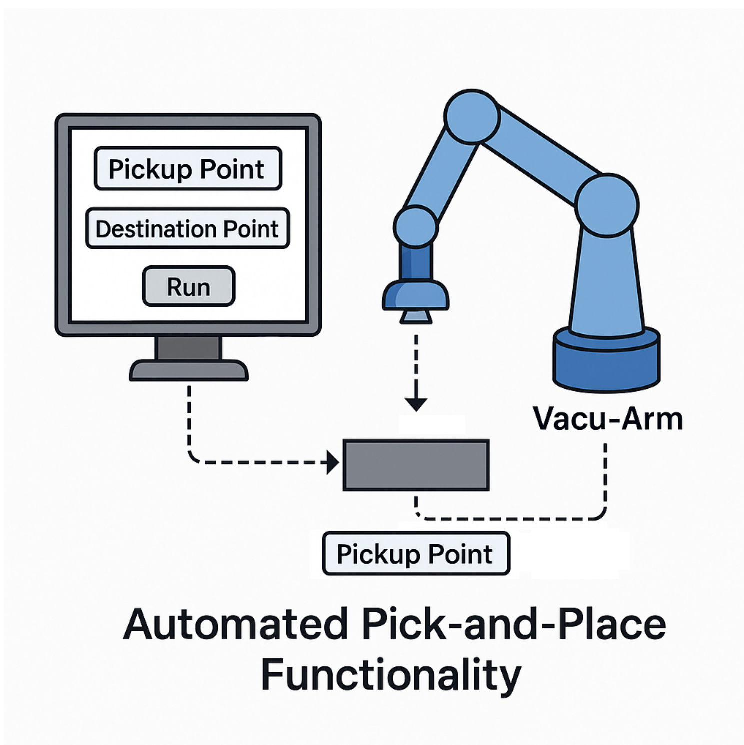

Figure 1: Vacuum Arm conceptual drawing

2 Product Description

The Vacu-Arm is a robotic arm that uses suction to pick up and move objects, making it ideal for repetitive tasks in workplaces like factories and warehouses. It helps reduce physical strain on workers, speeds up processes, and keeps performance consistent. The arm can run automatically or be manually controlled when needed, with a built-in camera to guide its movements. A second safety camera watches the area around it and stops the arm if it detects anything in the way, keeping people and equipment safe.

2.1 Features & Functions

The Vacu-Arm is built using the UR20 robotic arm, which is known for being fast, accurate, and easy to work with. It uses a vacuum-powered gripper at the end of the arm to pick up, move, and place objects. The system connects to the arm over a network using an IP address, which lets it send and receive commands remotely. The Vacu-Arm can run tasks automatically or be controlled manually when needed, giving users flexibility for different situations. To handle visual input, the system uses a separate microcontroller. This microcontroller processes data from two camera systems. The first camera is mounted near the gripper and helps the Vacu-Arm see and accurately move objects. The second camera is part of a safety system that watches the area around the robot. If a person or object gets too close, the microcontroller immediately stops the arm from moving to prevent accidents. The Vacu-Arm setup includes the UR20 robot arm, the vacuum gripper, the two cameras, and the external microcontroller, all connected to each other. A control panel or screen is also included for the user to set up or adjust the system.

2.2 External Inputs & Outputs

This subsection describes the key data exchanged between the Vacu-Arm system and external users or systems. The user can control the machine in two ways: by setting an automated task that specifies a pickup point and a destination point, or by manually controlling the arm using a camera system that provides visual feedback for precise movement and object handling. The table below summarizes the critical data elements involved in these processes, along with their descriptions and roles as inputs or outputs. Refer to Table 2

Table 2: Critical External Data Flows

Data Name	Description	Use (Input/Output)
User Command Input	Instructions from operator via control panel or interface, including task setup (pickup and destination points) or manual control commands via camera.	Input from end user
Task Configuration Data	Automated task parameters such as pickup location, destination location, and timing details.	Input from external system or user
Network Control Signal	Command messages sent over IP to control the UR20 robotic arm (e.g., move to position, activate gripper).	Output to robotic arm
Camera Feed (Primary)	Real-time video or images from the control camera mounted near the gripper, used to guide manual movement and object handling.	Input to microcontroller
Camera Feed (Safety)	Continuous video data from safety camera monitoring for obstacles or humans in the workspace.	Input to microcontroller
Safety Interrupt Signal	Signal from the microcontroller to immediately stop the robotic arm if an obstacle or person is detected nearby.	Output to robotic arm/controller
Status Feedback	System updates including task progress, error codes, and overall system status sent to the user interface.	Output to end user
Operation Logs	Records of movements, errors, and task completions for monitoring and maintenance.	Output to external system or user

2.3 Product Interfaces

The Vacu-Arm provides several operational interfaces tailored to different types of users, including end-users or operators, administrators, and maintainers. The primary interface for operators is a touchscreen control panel located near the base of the robotic arm. This panel displays a simple, intuitive graphical user interface that allows operators to initiate automated tasks by specifying pickup and destination points, or switch to manual control mode. In manual mode, the interface provides a live video feed from the primary camera mounted near the gripper, enabling operators to precisely guide the arm's movement and activate the vacuum gripper in real time.

The control panel also presents status updates, such as current task progress, system health, and any active safety alerts. Operators receive immediate notifications if the safety camera system detects an obstacle or person, prompting the arm to halt. Buttons for starting, pausing, and stopping tasks are clearly displayed, along with emergency stop controls for quick intervention.

Administrators access a more advanced interface, which may be hosted on a connected computer or tablet, to configure system parameters, manage user permissions, and update task routines. This interface includes dashboards that log operation history and error reports, assisting in system optimization and troubleshooting.

Maintainers interact with the Vacu-Arm via service panels on the hardware itself, which allow physical access to components like the vacuum pump and wiring. They also use diagnostic tools accessible through the administrator interface to monitor system performance, run calibration routines, and perform software updates.

3 Customer Requirements

This section outlines the customer requirements for the UR20 robotic arm to palletize uniform sized boxes. The project also includes the design and implementation of a custom robotic gripper that incorporates bellow suction cups to handle the boxes effectivelty. The robot arm will maintain full functionality in waiting for a box on the conveyor belt and locating it, picking it up and placing on a pallet.

3.1 Box Palletizing by UR20 Robot

3.1.1 Description

The UR20 will be used to palletize boxes, process data, and stack boxes in an effective manner. The system will handle boxes of the same size, ensuring precise and efficient stacking.

3.1.2 Source

CSE Senior Design project specifications

3.1.3 Constraints

Environment is suitable for operation of collaborative robot in a safe manner.

3.1.4 Standards

ISO 10218-1 Robots and robotic devices - Safety requirements for industrial robots ensuring protective measures. ISO 10218-2, Robots for industrial environments - Safety requirements to minimize hazards associated with robots and end effectors.

3.1.5 Priority

Critical The priority of this requirement relative to other specified requirements. Use the following priorities:

3.2 Custom Gripper with Vacuum and Bellow Cups

3.2.1 Description

The gripper design will feature bellow cups and a vacuum to securely handle and move the boxes during the palletizing process. It will be designed to fit the size of the boxes and ensure firm grasping to prevent slippage during handling. The vacuum will be generated using an air compressor and an attachment along the tubing line that utilizes the Bernoulli principle.

3.2.2 Source

CSE Senior Design Project specifications

3.2.3 Constraints

The strength and durability of the bellow cups that support the box's weight, and the selection of materials on a limited budget.

3.2.4 Standards

ISO/TR 20218-1:2018 Robotics - Safety design for industrial robot systems, ensuring safety measures for design and integration of end-effectors. ISO 10218-2, Robots for industrial environments - Safety requirements to minimize hazards associated with robots and end effectors.

3.2.5 Priority

High

4 Packaging Requirements

The packaging requirements for the UR20 robot and custom-designed gripper will include pre-installed control software. Additionally, the custom gripper will be fully assembled and contained in a single package. The requirements focus on making the product easily deployable and ready for use in palletizing operations.

4.1 Pre-Installed Software

4.1.1 Description

The control software for operating the UR20 robot will be pre-installed on the the system before delivery to the customer. The robot can immmediately perform palletizing application upon arrival without complex installation procedures.

4.1.2 Source

CSE Senior Design Project Specifications

4.1.3 Constraints

The software must be compatible with the UR20 system and ensure reliable functionality. Limited modifications will be needed by customer post-installation.

4.1.4 Standards

ISO/IEC/IEEE 12207:2017 Systems and software engineering - Software life cycle processes - proper installation and support for future updates

4.1.5 Priority

High

4.2 Gripper Assembly and Packaging

4.2.1 Description

The gripper mechanism, including vacuum suction cups, will be delivered pre-assembled and tested for immediate use with the UR20 robot. All components will be securely packaged to avoid damage during transportation.

4.2.2 Source

CSE Senior Design Project Specifications

4.2.3 Constraints

The packaging must ensure the safe transport of the assembled gripper, sourcing of packaging material will be the responsibility of the engineering team.

4.2.4 Standards

ISO 3676:2012 Packaging - Complete, filled transport packages and unit loads - Unit load dimensions ensuring products arrive safely from their origin to their destination.

4.2.5 Priority

High

5 Performance Requirements

Performance requirements for **The Vacu-arm** address operational speed, responsiveness, energy efficiency, and overall usability in time-constrained environments such as classrooms or labs. These requirements help ensure the system performs reliably under expected workloads and can be set up or operated efficiently. They also define expectations for motor responsiveness, system startup time, and power constraints, all of which are critical for maintaining user satisfaction and minimizing downtime.

5.1 Movement Response Time

5.1.1 Description

The Vacu-arm must respond to movement commands within 200 milliseconds of input, ensuring low-latency control and smooth operation during manual or automated tasks.

5.1.2 Source

User feedback and real-time control expectations

5.1.3 Constraints

Performance may vary depending on USB communication speed; all operations must remain under 250ms even under load.

5.1.4 Standards

ISO 9283 (Manipulating Industrial Robots - Performance Criteria)
ROS real-time communication guidelines (if used)

5.1.5 Priority

High

5.2 Startup Time

5.2.1 Description

The system must complete initialization and become operational within 10 seconds of being powered on or connected to a host computer.

5.2.2 Source

Usability best practices; team-defined usability targets

5.2.3 Constraints

Includes firmware boot-up, serial connection initialization, and readiness check.

5.2.4 Standards

IEEE 1471 (Software Architecture)
ISO/IEC 25010 (System Usability Characteristics)

5.2.5 Priority

Medium

5.3 Setup Time

5.3.1 Description

Initial setup (physical connection and software recognition) must take no longer than 5 minutes, assuming no prior technical knowledge from the user.

5.3.2 Source

Educational use case goals; instructor feedback

5.3.3 Constraints

Includes unpacking, connecting hardware, launching software, and verifying connection.

5.3.4 Standards

ISO 9241-110 (Interaction principles for usability)
IEEE 1063 (User documentation)

5.3.5 Priority

High

6 Safety Requirements

The Vacu-arm has to work safely in settings such as research laboratories, classrooms, and prototyping areas. This section explains the safety procedures that must be included in both the system's hardware and software to reduce user risk, avoid inadvertent operation, and meet applicable safety standards. These requirements are based on institutional safety regulations, industry standards, and team risk assessments completed during system development.

6.1 Laboratory equipment lockout/tagout (LOTO) procedures

6.1.1 Description

The Vacu-arm must contain a Lockout/Tagout (LOTO) protocol to ensure that the system may be securely turned off and deactivated during maintenance or troubleshooting.

6.1.2 Source

Industry safety protocols; lab safety manual recommendations

6.1.3 Constraints

LOTO must be possible without causing damage to sensitive components or necessitating the use of specialized tools.

6.1.4 Standards

Occupational Safety and Health Standards 1910.147 - The control of hazardous energy (lockout/tagout).

6.1.5 Priority

Critical

6.2 National Electric Code (NEC) wiring compliance

6.2.1 Description

The Vacu-arm must follow National Electric Code (NEC) criteria to ensure safe and dependable wiring throughout the system. Furthermore, electronic components must be protected against electrostatic discharge (ESD) during handling, operation, and maintenance to avoid damage or failure.

6.2.2 Source

CSE Senior Design laboratory policy

6.2.3 Constraints

Compliance should not interfere with system functionality or modularity. Wiring must be clearly labeled, inexpensive, and simple to install. High-voltage sources, as specified by NFPA 70, should be avoided whenever possible.

6.2.4 Standards

NFPA 70

6.2.5 Priority

Critical

6.3 RIA robotic manipulator safety standards

6.3.1 Description

An emergency stop button must be built into the Vacu-arm system to immediately stop all motion and vacuum power in the event of a malfunction or user hazard.

6.3.2 Source

CSE Senior Design laboratory policy, Team risk analysis, Mechanical safety norms

6.3.3 Constraints

Must be easily accessible, clearly labeled, and mechanically isolated from other controls

6.3.4 Standards

ANSI/RIA R15.06-2012 American National Standard for Industrial Robots and Robot Systems, RIA TR15.606-2016 Collaborative Robots

6.3.5 Priority

Critical

7 Maintenance & Support Requirements

The Vacu-arm will require ongoing maintenance and support to ensure its continued operation in classroom, research, and prototyping environments. Maintenance and support responsibilities include error correction, hardware replacement, troubleshooting support, and access to necessary documentation and source code. These requirements are defined with the assumption that either internal technical staff or external customers may need to support **The Vacu-arm** "in the field." To that end, documentation, tooling, and source code accessibility are critical to sustaining system performance after initial deployment.

7.1 Access to Source Code and Documentation

7.1.1 Description

Full source code and technical documentation for **The Vacu-arm**, including build instructions, interface specifications, and system architecture, must be provided. This ensures maintainers can debug, rebuild, or enhance the system when necessary.

7.1.2 Source

Open-source practices, internal maintenance team requirements

7.1.3 Constraints

All documentation must be in human-readable format (e.g., Markdown, PDF) and stored in a version-controlled repository (e.g., GitHub or GitLab).

7.1.4 Standards

IEEE 828 (Configuration Management)
ISO/IEC 26514 (Documentation for software users)

7.1.5 Priority

High

7.2 Troubleshooting and Maintenance Guides

7.2.1 Description

Detailed maintenance and troubleshooting manuals must be developed for both hardware and software subsystems of **The Vacu-arm**. These guides should include diagnostics steps, error codes, and common failure resolutions.

7.2.2 Source

User experience expectations; internal QA policy

7.2.3 Constraints

Documentation must be understandable by users with limited robotics experience; visuals or diagrams should accompany textual instructions.

7.2.4 Standards

ISO/IEC 26513 (System and software engineering – Requirements for testers)
ISO 9241-110 (User interfaces)

7.2.5 Priority

High

7.3 Tooling Requirements for Maintenance

7.3.1 Description

All specialized tools (hardware or software) required to maintain **The Vacu-arm** must be documented and, where possible, distributed with the product or made easily accessible. These include specific calibration tools, drivers, or interface software.

7.3.2 Source

Hardware team input; sprint testing observations

7.3.3 Constraints

Avoid reliance on proprietary tools unless necessary; prioritize open-source or low-cost maintenance solutions.

7.3.4 Standards

ISO/IEC 19770 (Software Asset Management)
General ESD (Electrostatic Discharge) handling best practices

7.3.5 Priority

Medium

7.4 Support Environment Specifications

7.4.1 Description

A clear specification of the software environment required to run diagnostics and maintenance tasks on **The Vacu-arm** must be provided. This includes supported OS platforms, Python versions, and necessary libraries.

7.4.2 Source

Software team input; platform compatibility goals

7.4.3 Constraints

Environment must be cross-platform (Windows, Linux, macOS) and easily reproducible with documented setup steps or environment files (e.g., `requirements.txt`).

7.4.4 Standards

PEP 508 (Python dependency specification)
Docker or virtual environment usage standards (if applicable)

7.4.5 Priority

Moderate

8 Other Requirements

The Vacu-arm is a programmable vacuum-based robotic arm system designed for precision material handling in research, educational, and prototyping environments. To ensure the system meets user expectations and performs reliably in a range of scenarios, this section outlines additional requirements not previously covered. These include configuration steps for customer setup, platform compatibility, software modularity, and architectural guidelines to support future enhancements. Meeting these requirements is essential for the successful deployment and long-term scalability of **The Vacu-arm**.

8.1 Modular Design

8.1.1 Description

The Vacu-arm must follow a modular software and hardware design, allowing individual components (e.g., vacuum controller, movement controller, sensor feedback) to be updated, replaced, or extended independently without requiring major rework of the entire system.

8.1.2 Source

Engineering best practices, team decision during Sprint 2.

8.1.3 Constraints

All modules must follow a defined API/communication protocol to ensure compatibility. Any changes to one module must not break the functionality of others.

8.1.4 Standards

IEEE 1471 (Software Architecture)
ISO/IEC 42010
ROS modularity guidelines

8.1.5 Priority

High

8.2 Extensibility for Future Enhancements

8.2.1 Description

The Vacu-arm must be designed with extensibility in mind, allowing new features such as additional sensors, AI-based decision-making, or alternate gripper types to be integrated with minimal refactoring.

8.2.2 Source

Team foresight, advisor recommendation

8.2.3 Constraints

New features must integrate using the existing control and communication structure. Extensions must be backward-compatible.

8.2.4 Standards

SOLID principles (Software design)
IEEE 830 (Software Requirements Specification)

8.2.5 Priority

Medium

8.3 Portability Across Platforms

8.3.1 Description

The software controlling **The Vacu-arm** will only be conrtolled via URscript, which is integrated into the PLC with the 3PE pendandt device.

8.3.2 Source

User requirement; educational use across diverse lab setups.

8.3.3 Constraints

Use only cross-platform libraries (e.g., Python standard library, PySerial for communication). Avoid OS-specific commands or tools.

8.3.4 Standards

POSIX compliance where applicable
Python PEP 8 for cross-platform scripting
USB/serial communication standards

8.3.5 Priority

Medium

8.4 Customer Setup and Configuration

8.4.1 Description

The final product must include a simple configuration guide and scripts to enable users to set up **The Vacu-arm** with minimal technical background. The system should support plug-and-play recognition when connected to a host computer.

8.4.2 Source

Usability and accessibility goals; feedback from target users.

8.4.3 Constraints

Setup process must not require manual driver installation or low-level configuration changes. Must work with default Python installation and common USB interfaces.

8.4.4 Standards

ISO 9241-210 (Ergonomics of Human-System Interaction)
USB HID standards (if applicable)

8.4.5 Priority

High

9 Future Items

In this section, we summarize requirements that were discussed during development and assigned a **Priority 5 (Future)**. These items are considered valuable enhancements to **The Vacu-arm** system but are outside the scope of the current prototype due to constraints such as time, budget, or technical limitations. The items listed here are duplicates of entries from Sections 3 through 8.

9.1 Advanced Voice and Gesture Control Interface

9.1.1 Description

The system will support real-time voice and hand gesture control to allow for contactless operation of the Vacu-arm. Operators will be able to speak commands or use predefined gestures (e.g., closed fist to grip, open hand to release) recognized by a depth-sensing camera.

9.1.2 Source

User feedback; industrial control trends; educational goals

9.1.3 Constraints

Accurate speech and gesture recognition requires robust AI models and error handling to prevent unsafe actions.

9.1.4 Standards

ISO 10218-1 and ISO 10218-2 (Industrial robot safety)
ISO/TS 15066 (Collaborative robot safety)

9.1.5 Priority

Future

9.2 Mobile Power Integration

9.2.1 Description

Future versions of the Vacu-arm may include a battery pack or portable power supply to support mobile or off-grid operation.

9.2.2 Source

Team roadmap; mobile use case exploration

9.2.3 Constraints

Battery weight, safety, and power capacity must be balanced for real-world tasks.

9.2.4 Standards

IEC 61960 (Battery performance testing)
ISO 26262 (Functional safety for electrical/electronic systems)

9.2.5 Priority

Future

9.3 Remote Web Interface for Monitoring

9.3.1 Description

The system may include a remote-access dashboard for monitoring Vacu-arm task performance, system health, and video feeds via browser.

9.3.2 Source

Future system extensibility planning

9.3.3 Constraints

Requires secure IP-based access, user authentication, and live data streaming support.

9.3.4 Standards

OWASP Web Security Standards
ISO/IEC 27001 (Information Security Management)

9.3.5 Priority

Future

References