

Project Details	Competencies Provided
-----------------	-----------------------

- ◆ **Deliverables:** Wireframes, Prototypes, Prioritized Backlog

- ◆ **Methodology:** Customized Agile
- ◆ **Tools:** Figma, Notion, Mural, Excel, GitHub, VS Code
- ◆ **Software Architecture:** VUE.js, C++

Interim Software Leadership

by a kitchen-staff-friendly UI with enhanced visual hierarchy and emphasis to get the right information onto each page, organized by importance and user role.

through a single documentation repository and backlog for both design and development. Created and prioritized user stories to seed a feature roadmap.

worried and actionable error messages. This meant providing errors in the right way, at the right time, using audible chimes when necessary.

formalized testing environment. In other words, UI changes could only be tested once deployed, so extra care had to be taken to overcome inefficiencies.

We implemented a lightweight version of Scrum with a weekly checkpoint for design review and backlog grooming, a daily Slack stand-up channel in lieu of meetings, and monthly code reviews. Formalized user stories were written very early on in our process to drive development tasks. We started the project by asking ourselves several key questions:

- ◆ What other systems do the end users and the UI need to interface with?

- ◆ How much should our system match the look and feel of the other systems it will integrate with? Customization versus the principle of least surprise.
- ◆ How much control will the UI grant to the end user to modify the preprogrammed workflow of the robot? How much trust will we place in the end user?
- ◆ How will we differentiate a component that has a fixable error (i.e. validation) from a component that is broken?
- ◆ Can AI be used to suggest error correction strategies or to correct errors automatically?
- ◆ How do we create an experience that is so effortless that the end users prefer to control the robot through the UI, instead of resorting to guard pulls or MStops?

sary func
messagin

Other Deliverables

_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

* Details have been abstracted to meet client confidentiality requirements.

Lemma 5.1. *If $\mathcal{F} \in \mathcal{M}_n$ is a $\mathcal{P}$ -maximal family, then $\mathcal{F} \in \mathcal{M}_n$ is a $\mathcal{P}$ -maximal family if and only if $\mathcal{F} \in \mathcal{M}_n$ is a $\mathcal{P}$ -maximal family.*

SITE								
------	--	--	--	--	--	--	--	--

SYSTEM LEVEL

STATION LEVEL

COMPONENT LEVEL

* Details have been abstracted to meet client confidentiality requirements.

Wireframes / UI Updates

- ◆ Present more relevant information for each table row including production status, time completion, and position.
- ◆ Move alerts to be “in-line” instead of only available on a global alerts reporting page.

- ◆ Make information easier to cross reference across various sections and pages in the UI.
 - ◆ Remove jargon and complexity where possible to optimize UI for operational staff - not robotics engineers.
 - ◆ Use more visually appealing and functional components, implementing fly-outs and removing undersized modals.
 - ◆ Standardize button colors and sizes.
 - ◆ Add functionality for customization that didn't currently exist, i.e. order modifications.
 - ◆ Use design elements in addition to text to get users at-a-glance information on order and robot status, fill levels, etc.
 - ◆ Group alerts by type and severity to avoid information overload and to allow the user to triage next steps.
 - ◆ Introduce control panel for audible alerts so that the end user has some control over sound effects, enabling them to minimize audible alerts from being either too minimal or too distracting.
- 



High Fidelity Screens / Prototyping

High fidelity screens were created in Figma and linked together into a clickable prototype. This

The diagram illustrates the transition from a table view to a list view. On the left, a tablet displays a table with columns for 'Name', 'Age', 'Gender', and 'Status'. The table contains three rows of data. On the right, a smartphone displays a list view of the same data, with each row represented as a list item with a red background. A curved arrow points from the tablet to the smartphone, indicating the transition.

