



QPoint

Robotic Solutions

Web Based User Interface

Menu Structure

GOAL

- Replace the current code based menu structure with a web page based menu that allows
 - cleaner operator interaction
 - more attractive, and current appearance
 - Full functionality with main supervisory application
 - Requires ZERO connectivity with outside world (local machine operation, no www access)
- Menu structure is likely available from various code examples, don't want to pay any ongoing royalties, but one time purchases are not out of the question for a code base that is useful
- Web page must send and receive messages from supervisory application for data, and control response using java script. Examples provided
- System will need to run OFFLINE on a windows based PC, .Net container is CHROME based

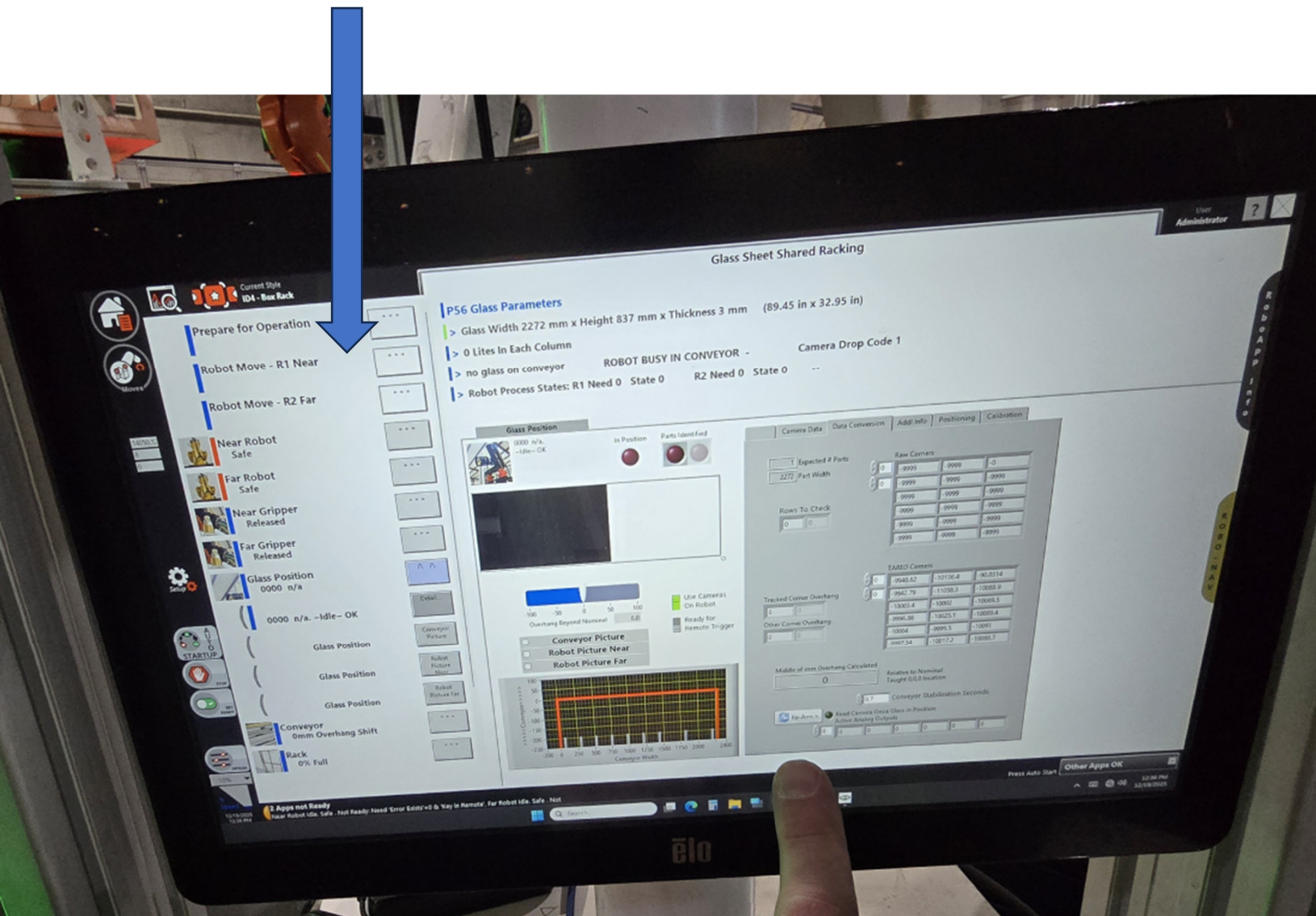
Any robot + Our products = **INSTANT AUTOMATION**

QPoint
Robotic Solutions

We didn't make the robot...
We made the robot EASY.

Current Functionality

- Left hand menu structure



The screenshot displays the P56 Glass Parameters software interface. On the left, a sidebar contains a 'Moves' list with items like 'Prepare for Operation', 'Robot Move - R1 Near', 'Robot Move - R2 Far', and 'Enter Auto Mode'. The central status area shows 'P56 Glass Parameters' and 'No Glass Size Specified'. The main window features a robot position graph and a 'Far Robot' status panel. A yellow callout box highlights a 'Poor Communication Excessive Time-Outs' error message.

Moves List:

- Prepare for Operation
- Robot Move - R1 Near
- Robot Move - R2 Far
- Robot Management
- Enter Auto Mode
- Near Robot SafeNot Found:
- Far Robot SafeNot Found:
- Near Gripper Released
- Far Gripper Released
- Glass Position 0000 n/a
- Conveyor 0mm Overhang Shift
- Rack 2% Full
- Cullet Bins
- Turn Table Not OK

P56 Glass Parameters

- > No Glass Size Specified
- > Gripper HELD RELEASED
- > R1 Need 0 State 0 R2 Need 0 State 0 --
- > Drop Code -9 Glass Ready 0 >> Clear of Conveyor 0

Far Robot

Not Found
ZERO mm to go
SafeNot Found: .
OK

Current Position (X,Y,Z)
(Rx,Ry,Rz)
Additional Axis

Joint Angles OK

mm/Sec

deg/sec

Poor Communication Excessive Time-Outs

Moves

- Prepare for Operation
- Robot Move - R1 Near
- Robot Move - R2 Far

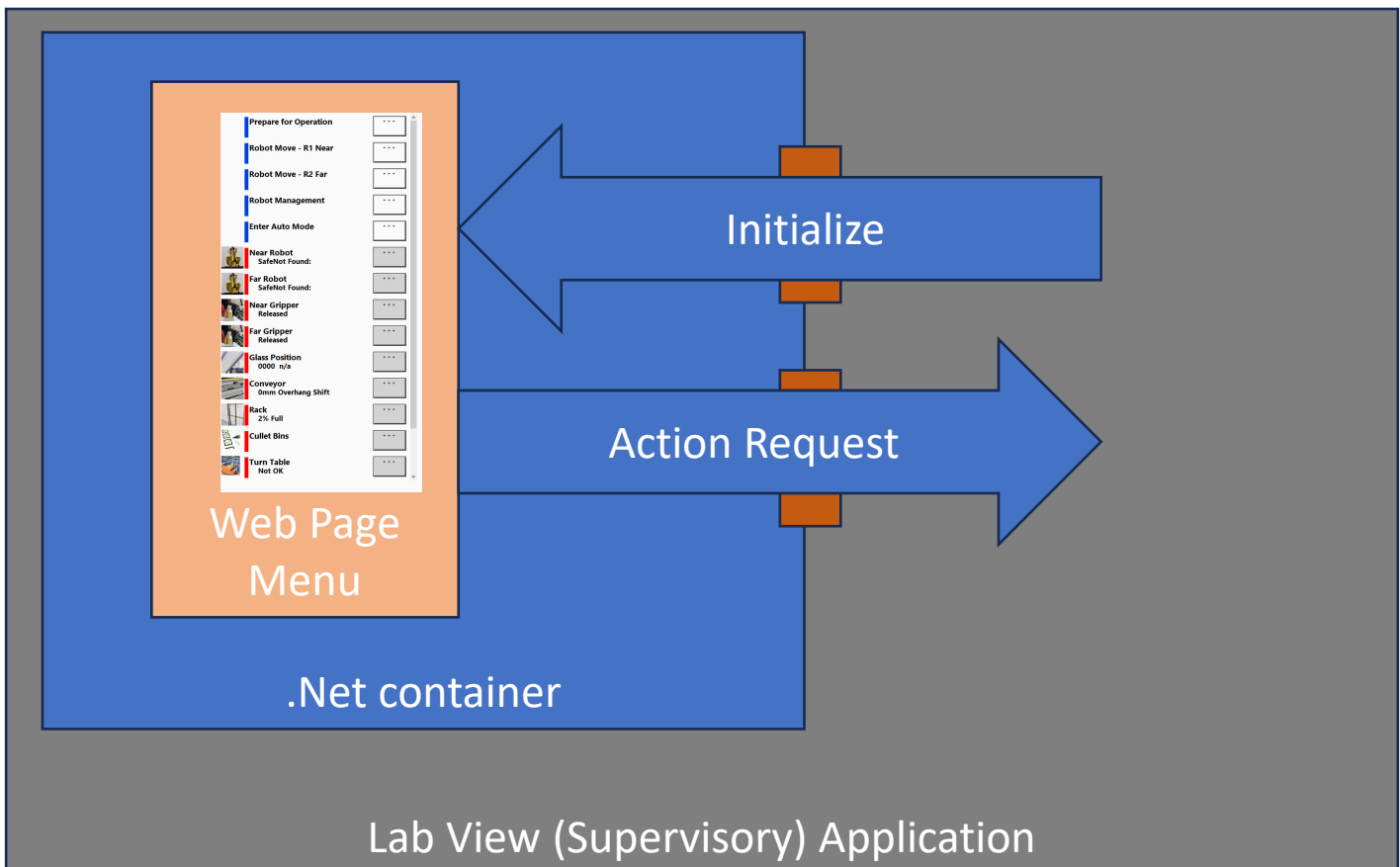
The image displays the QPoint Robotic Control System interface, which is divided into several sections. On the left, a vertical sidebar contains icons for Home, Move, Setup, and Auto Start/Stop. The main area is titled "Menu Structure" and lists various robot management tasks. A large yellow arrow points from the "Menu Structure" title to the list of tasks. The tasks are as follows:

- Robot Management
 - Enter Auto Mode
 - Near Robot SafeNot Found:
 - Far Robot SafeNot Found:
 - Near Gripper Released
 - Far Gripper Released
 - Glass Position 0000 n/a
 - Conveyor 0mm Overhang Shift
 - Rack 2% Full
 - Cullet Bins
 - Turn Table Not OK

The right side of the interface shows a detailed view of the "Far Robot" status, including its current position (X, Y, Z), joint angles, and speed. It also displays a "Setup" button and a "STARTUP" button. The bottom of the screen shows a Windows taskbar with various application icons.

Technical Structure

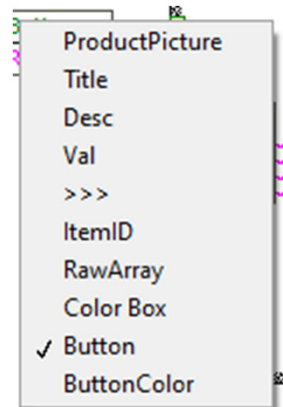
- The web page will reside inside of a .Net container in a LabView (windows based) application.
- The web page will reference a WebView2 code base, running a visual studio developed .Net application to house the interaction handling between the webpage and the LabView application.
- The .Net interactions are tested
- Required focus in on the web page, and establishing the rules for the interaction subroutines
- ***NO CODE*** is needed to be developed for the .Net, or LabView portion. We will handle anything needed for that.
- Sample web page provided



Vertical Menu

Elements for an item Appearance

- Heading
- Sub Text
- Button
 - Action Type (designates coloring/appearance)
 - Menu Expansion, Detail view of Item, Device control, Robotic movement
 - Button text
 - Enabled
- Image



Page Size

- 460 pixels wide (approx.)
- Should have room for at least 10 rows in vertical of 874 pixels or so
- Vertical scrolling ok, NO horizontal scrolling
- Menu should show ONLY active items in category, not a full expanding list (like a phone menu where only the active group is shown, and a back or up button to return to the previous category)

Any robot + Our products = **INSTANT AUTOMATION**

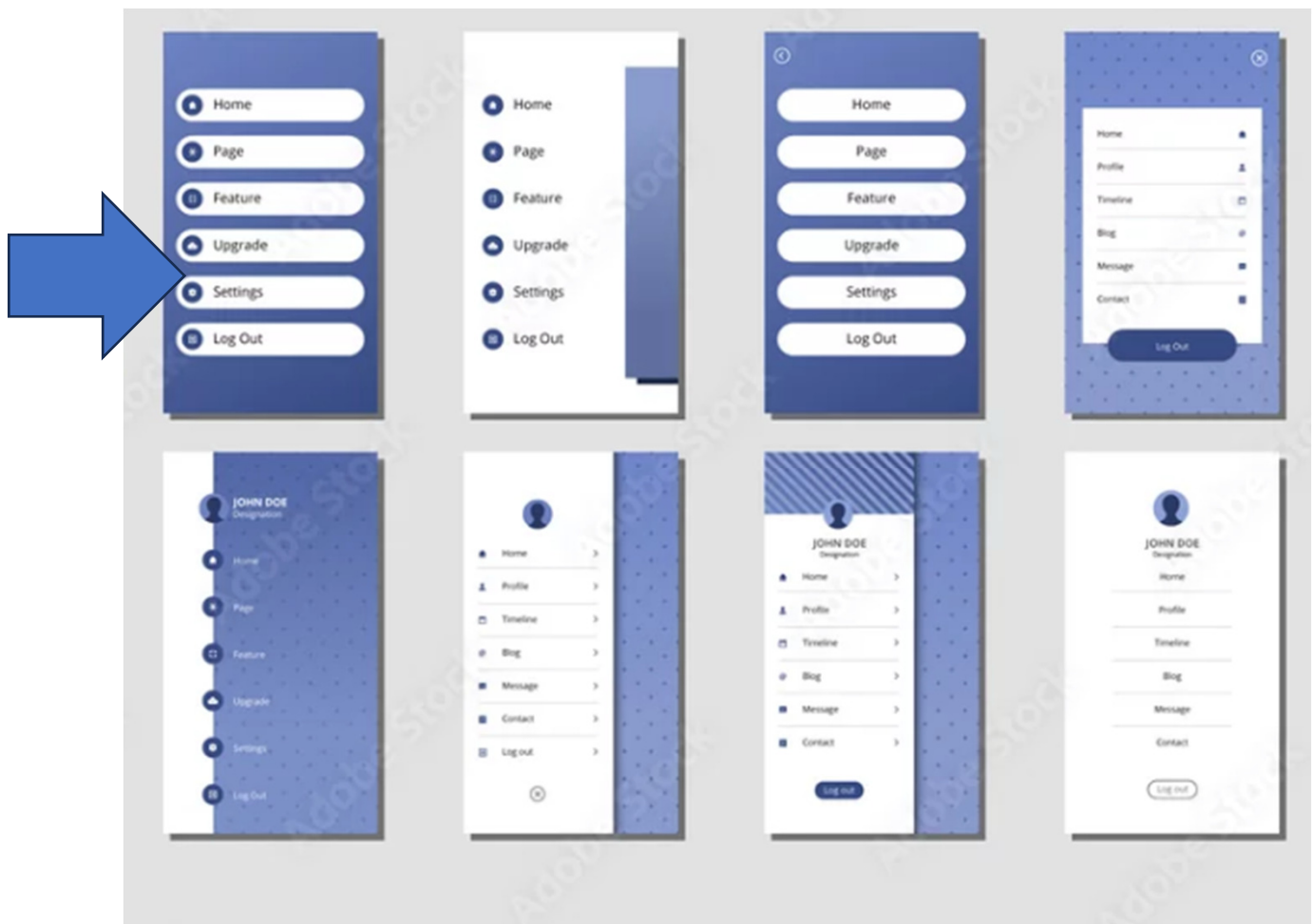
QPoint

Robotic Solutions

We didn't make the robot...
We made the robot EASY.

Appearance

- Rounded buttons as upper left is consistent with theme of application



Interaction with Supervisory Application

From supervisory Application

- Initialize display
 - Sends info for list build
 - Can be multiple calls, or whatever is needed to send full menu structure info
- Update display
 - Supervisory application will have live data and live status info for each item in the menu structure which will need to be refreshed (button enables based on status, change in subtext info, etc.)

To Supervisory Application

- Button presses to trigger actions
- Active sub menu (to know which nodes need to be updated for live data)

Related Code Snipents

- `webPageActivity(string);`
 - Sends a string to the supervisory application when something is clicked
 - This needs to be incorporated in your menu java code

```
circle.style.cursor = "pointer";  
circle.addEventListener("click", () => {  
    webPageActivity(node.name); //*****  
});
```

- We can call (trigger) from our supervisory application any java script function. Easiest if they receive a single string as the parameter, but can send multiple parameters too as needed.
- These will be for initialization and updates to the live display data

```
function getGreeting(name) {  
    return `back at ya, ${name}!`; /***we can and need return functions for some, we see this data  
}  
  
function message(messageTxt) {  
    alert(`script called with parameter: ${messageTxt}`);  
}  
  
function setBackground(color) { // will likely want some visual adjustment style functions to be available  
    document.body.style.backgroundColor = color;  
    alert(`script called with parameter: ${color}`);  
}
```