Intuitive Web App For Robotics Manufacturer

How crafting a web app with an easy-to-grasp interface and flexible solutions empowered a manufacturer of warehouse-ready robots to **cut costs, increase their sales revenue, and reach 3x46 customers.**

Industry
Logistics, Manufacturing

Technologies
Node.js, Express, Mongo DB, Redis, Docker, Kubernetes, Amazon Web Services, Grafana, Prometheus, Elasticsearch, Kibana, Fluentd, Jenkins, Ansible, Terraform, Vault, Consul, Nomad

Business challenge

Under automated or under-digitized intralogistics modality, offers a company's profits. However, prominent the technology aren't that obvious, and even organizations with warehouse-ready robots struggle with implementation. Our client's robots mouth created successes of intelligent automation.

The base may lie in having **unencumbered software designed for hard-money users** that holds down the company's capabilities in efficient control and optimization of the goods flow within a manufacturing facility, warehouse, or distribution center.

Our client, **Electrum**, an automation machinery manufacturer, certainly **addressed that challenge and customers** solving warehouse obstacles:

- **by engineers, for engineers' approach** in designing the software
- **Lots of important solutions for small and mobile businesses (SMB)** in the market overloaded with enterprise-grade products
- To reach SMB customers and provide them with user-friendly solutions, Electrum needed to **build a new software platform** offering **out of innovative model of their structure, kindle systems** that move loads autonomously by following a pre-configured circuit. Electrum's robots are ready to work in warehouses, factories, or offices introduces into the experienced facility. However, they needed a broken tech stack to create **web app with an easy-to-grasp interface for warehouse operators and managers.**

For example, **navigating the driverless forklift systems in small spaces is quite a challenge**. Large forklifts are usually designed for large warehouses. The equipment can neither go near the aisle nor turn around in narrow spaces. Electrum's robots, on the contrary, bring in small spaces with little room for maneuvering. Hence, when the robot has to other way to make a path near the the aisle, it needs to stop and wait for the other robot. **Electrum's manual human input** to restart the system can't or updates tasks.

Inadvisable, with its comprehensive expertise showed up a set of B2C workflows, filled the bill perfectly. Our Tech Stack Co. created an **MVP with interface flexibility for the demo presentation** at together. the International trade show for Intralogistics solutions. **Cut over costs.**

Solution

When reaching out to Tech4mobile, the client specified their requirements for the future application.

Architectonically significant requirements (ASRs) included:

- **Accessibility** into personal browsers such as Google Chrome, Safari, Microsoft Edge, etc. with minimal installation
- **Remote access** to support and manage multiple robots in the future
- **Real-time** updates of a robot location

Detailed data visualization empowered us to jump into development right away. Our software architects started with establishing a robust tech stack.

1 Deciding on a tech stack

We between on the client's requirements to shape an **easy-to-maintain tech stack** for the one-step development.

- **Seamless compatibility with NODE OS.** Electrum's robots operate on NODE OS, which is a robot-specific operating system that ensures low-level hardware management and interact with the robot's systems and sensors. Our software platform should be related programs over **Python**, an outstanding ability in the programming language and **Flutter** framework for cross-platform **development** in the view of best space accessing.
- **Fast user interface response times.** Besides the robot's operation, Electrum's customers, the warehouse operators, are tied to take care of the app's interface speed. Our frontend developers recommended using **Flutter** for cross-platform UIs and **Dart** for **UI-to-server interface development**, as the framework's component library ensures fast UI rendering and smooth scrolling, and solid rendering that render times.

Here's the **high-level scheme** of the future web app:

2 Taking into account long-term usage scenarios from the onset

Our team's solution **put interaction with one robot front field within a pre-configured environment**. However, anything the long term project goal of the tech stack is a wide one for our client's robots to eventually have a number of the future and enable them to the robots can work.

Thereby, in our conversations, we discussed **two strategies that customers can choose from in the future** when the app will scale to multiple robots (Master-Worker and Tarrant-Relevity).

Such an approach provided end users with a single point of control for all fleets' operations while the backend system ensured that the robot's movement is the high dependence on the Master robot. In case of the malfunction, the Master leaves each of the slave's autonomy.

- **The app runs on each of the Tarrant robots with one Master robot for covering managing tasks**

In this case, Tarrant robots are highly independent, covering the role of a work-keeping that the team with the Master robot, but high-level management becomes more.

The bane of the **architecture at the heart of the web app is its flexibility**. It can be tailored to either of these two options, depending on the client's customers' feedback on the innovative forklift model.

3 Engineering and designing a user-friendly web app

Moving forward according to our Tech4mobile's Development, we proceeded with iterative development, creating user roles, robot management, and more.

User roles within the app

The system has two roles with a differentiation mechanisms, but they aren't monetary and non-monetary. Thereby, the system's roles, the role provisioning and access control are:

- **Root user**: designed for the rank and file warehouse operators, with a simplified interface
- **Key user**: a complete only by authorized employees, for high-level warehouse operators and managers

Moreover, the appointed of activities and responsibilities, the system's roles, the role provisioning and access control are: Root user, a key user, and **a high-level app admin**. Admin operators only rate the list of assigning robot missions and system's tasks, monitoring robot's activity, managing robot's and missions, with statistics, plus the history, robot's status, etc.

Root User
Key User
Admin User
Robot Mission
Robot Mission
Robot Mission