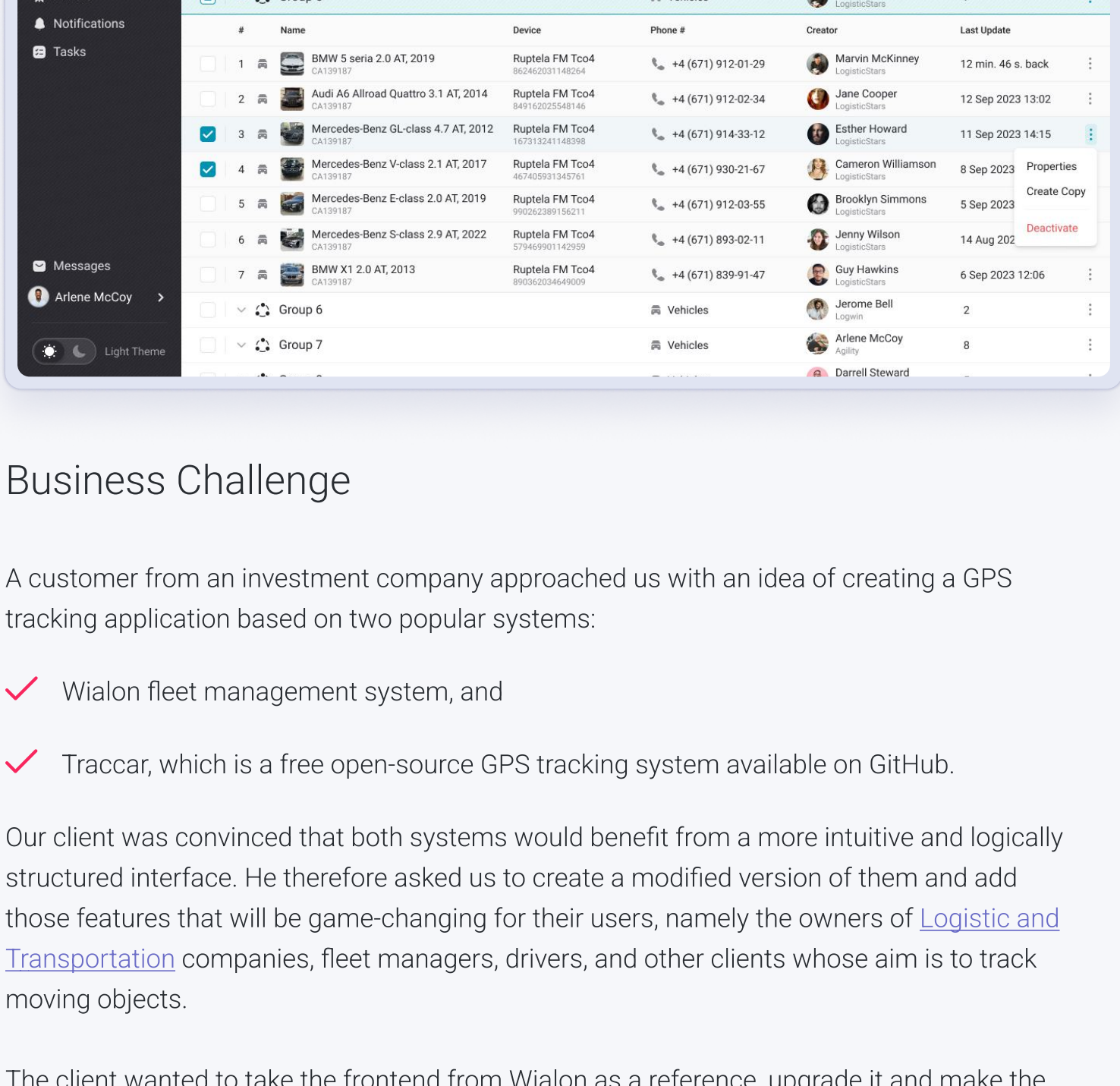


Traccar-Based GPS Tracking System

A GPS tracking system for commercial moving companies and other businesses that deal with monitoring, transportation, and relocation of different objects, including office furniture, lab equipment, animals, vehicles, warehouse items, etc., equipped with special sensors.



Business Challenge

A customer from an investment company approached us with an idea of creating a GPS tracking application based on two popular systems:

- ✓ Wialon fleet management system, and
- ✓ Traccar, which is a free open-source GPS tracking system available on GitHub.

Our client was convinced that both systems would benefit from a more intuitive and logically structured interface. He therefore asked us to create a modified version of them and add those features that will be game-changing for their users, namely the owners of [Logistic and Transportation](#) companies, fleet managers, drivers, and other clients whose aim is to track moving objects.

The client wanted to take the frontend from Wialon as a reference, upgrade it and make the application operate on the backend from Traccar. They liked the server given by Traccar and wanted Traccar customization. Besides that, more than 140 companies in the world use Traccar, so it was a decision not meant for discussion.

Our client's observations provided a valuable starting point for our analysis of these systems and helped us formulate a vision for future improvements in line with their requirements. Here's what was gleaned from the joint communications session:

- ✓ Both systems have been around a long time. They got better by adding new features, but some of these features were added on top of the existing architecture and logic, making the system heavier and the user experience less comfortable. For example, the reporting feature was scattered across the entire app, whereas it would be more efficient to gather all reporting tasks in one place.
- ✓ The system's design needs to evolve to keep up with modern UI/UX standards. However, it's crucial to maintain the similar functionality of the controls and keep the user interface and user experience feeling familiar to former customers.

After thorough market analysis and customer requirements gathering, we proposed a solution: creating a new frontend (Wialon as a reference) and adapting it to the Traccar backend. This approach allows us to integrate best practices and create a logically improved architecture without rebuilding existing systems from scratch.

Solution

To grasp the full idea of our client and reach the outcome they expected, we worked in a close collaboration, constantly communicating and exchanging our opinions. The thorough analysis of the possibilities for frontend implementation and the Traccar backend customization helped our team to get the full picture and know what we would be working with.

Traccar Customization

To know how to successfully implement all the changes we planned on frontend and adapt them to backend, our developers and business analysts investigated the existing **Traccar** backend. During the development process, we had to adapt some of the **Traccar** features to what we wanted to implement. Thus, for example, we had to match available fields from **Traccar** with the Wialon fields. In some cases, missing fields were created as additional custom fields.

Working with the Traccar backend gave us valuable experience, because there were also situations when we had to request the Traccar team to make some **changes** on the backend:

- ✓ Add an attribute "Administrative" for custom fields, so the creator could mark the fields with a relevant role (admin, manager, user) to let users control the fields they need from the backend;
- ✓ Traccar had an API for uploading images of the moving objects. We requested an API to store images of drivers and geofences;
- ✓ Add possibility to get the history of the driver assignments;
- ✓ API for list of the supported devices types;
- ✓ Add feature to return the driver along with the assigned device on demand;
- ✓ Implement possibility to receive the fuel level in the position attributes instead of the fuel consumption by period;
- ✓ Implement grouping for tables in report templates;
- ✓ Add an attribute "type" to Messages.

UI Changes

We clearly needed an application for tracking commercial trucks and analyzing information gathered from data devices about the health, speed, and positioning of the vehicle units, the level of technical liquids, etc. One of the key challenges that we were dealing with was the fact that the Wialon application is very rich in functionality but had an outdated and illogical frontend. The functionality of the future application was expected to be constantly updated and extended. [React JS](#) and [Webix](#) were considered as the main technologies.

Webix could be useful for quickly implementing additional functionality, as it has a large set of ready-made complex UI widgets, but it would require frontend development from scratch. And since **Traccar's** frontend is already built on React JS, the choice was obvious.

To improve the user interface, make it more intuitive, and add more clarity about users' roles and available features, we implemented **two modules**.

Monitoring Module

This is basically the side of an app that is accessible for all users depending on their roles. After logging into the user's application, all access rights come from the server to the frontend. Depending on these access rights and user role, only the content and functionality available for a particular user is displayed during the session.

The system for users with limited access now includes 3 tabs:

- ✓ **Tracking.** A user can monitor and manage Objects or be included in Groups to perform the tasks set for a group of users. Here, it's also possible to build quick reports for each appropriate Object or Group, and the system quickly redirects users to the Reports tab.
- ✓ **Geofences.** Users can check relevant Geofences or those that were included in certain Groups.
- ✓ **Reports.** We completely reworked this feature to add logic and intuitively to the reporting process. Users can now use hardcoded or custom Report templates to get information about Tracks, check Messages, and create or display other Reports for any Object/Group for any period. Some of the reporting features were available for Admins only, but we decided to transfer them to the User side to let drivers and other employees create better reports including all required data.

Admin and Super Admin Module

As for the administrator side of the application, it also includes more logically structured functionality now. Our customer wanted to create an administrative module within their application that would be visible to users with certain rights without the need to launch a separate application. Also, after transferring all reporting features to the Monitoring Module, it gave us more space to improve other features. As a result, the Administration side of the app got the following tabs:

- ✓ **Accounts.** An account is a unity that usually contains a user, data on their access rights, and resources (data on geofences, notifications, etc.).
- ✓ **Users.** A user is a person who logs into the system and has access to its objects in accordance with the granted rights.
- ✓ **Objects.** These are all the objects with implemented sensors that are tracked by the system (tracked vehicles, moving and stationary machinery, people, animals, etc.).
- ✓ **Drivers.** Drivers are also considered as system objects, however they have specific features that differ from other objects. Drivers don't use the app and don't have accounts but are able to use some company benefits (for example, to use cards to pay for fuel). All the data on the objects they are assigned to and their activities are tracked by the system.
- ✓ **Notifications.** The tab helps to create, monitor, enable, and disable notifications, whether they are required for particular users, groups, or objects. They help to register and control different events, including object movement, sensor readings, driver assignments, and so on.
- ✓ **Tasks.** This is the tab with all active, inactive, new, completed, and planned assignments that allows admin to have more control over operations.

Some **React pattern complex components** were implemented on frontend to be used across the application. Mostly in the Administrative module, we added **tree-tables** for grouped objects and drivers. All of the data tables and tree-tables were supplemented with the following **pattern functionality**:

- ✓ Multi-level sorting;
- ✓ Popup with configurable frontend filters for all columns, and the logic to configure a certain filter type depending on the value type in the column;
- ✓ Columns visibility setup;
- ✓ Checkboxes with specific interaction logic for bulk actions;
- ✓ Row sorting logic.

Traccar provided a functionality to transform raw data received from sensors. But we needed to provide users with a **converter** that has a familiar user interface for creating conversion rules. We also developed a **parsing panel** to let users preview row data in HTML or in data table view and parse them into the columns of the corresponding Report data table.

During the initial stage of the project, all components and their features were pre-designed in an integral style and added into a Storybook to be used across the application. Each new component was implemented the same way.

UX Changes

To improve user experience and decrease the amount of struggles, we added **Dynamic tables into the Reporting feature**. For managing dynamic data tables for Report templates, we created a **Data table wizard popup**. With its help, users with manager rights are now able to create, save, and update data tables. A user is now simply able to select data from various sources, group and sort data, and specify a diagram will be available for the chosen data table.

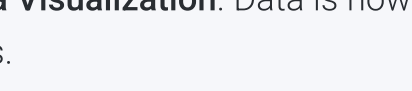
We have arranged the modules in a **more logical and intuitive hierarchy**. Messages and tracks, which are also reports with a hard-coded template, were located separately in the menu. After analyzing the system design, we created the same two templates for ourselves, but included them in the list of templates, marked as *"standard report"*, which cannot be edited. This helped users to work with the reports they are used to or copy them to create new templates.

The executed Reports are now displayed on the **Dynamic report page**, that contains three blocks interacting with each other: Map, Diagram, and the scope of data tables, displayed on the dynamic horizontal tabs. User actions on one of the blocks are reflected on the others.

Other Changes

The new application not only repeated the functionality familiar to Wialon, but also enriched it with new ones. To make the functionality consistent, we added the ability to make format conversion templates that could be saved and set up as visible for other users to reduce work time consumption.

Multi-language approach was also implemented on the new application architecture, so all text blocks on the frontend weren't hardcoded but taken from the file with the English version. It will allow the customer to add files to the system with any other languages they need.



12+

Month

Duration

2300+

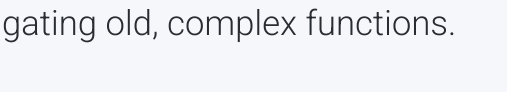
Estimated



man-hours

5

Team Size

Team Size



Results

The customer confirmed that the updated system has elevated his logistics to a new level, providing online monitoring of transport for any reporting period.

- ✓ **Convenience and Efficiency.** The updated application proved to be very convenient and effective for tracking commercial trucks and analyzing information.
- ✓ **Optimization and Customization.** The final result showed that the system can be optimized and customized according to the client's needs, providing a more logical user experience.
- ✓ **Clear Data Visualization.** Data is now clearly visualized, allowing users to better understand their tasks.
- ✓ **Time and Effort Reduction.** The new system helps reduce the time and effort previously spent navigating old, complex functions.

Your questions and requests are always welcome!

Visit

<https://xbsoftware.com/>

Contact Us

© XB Software - Software Development Company. All rights reserved