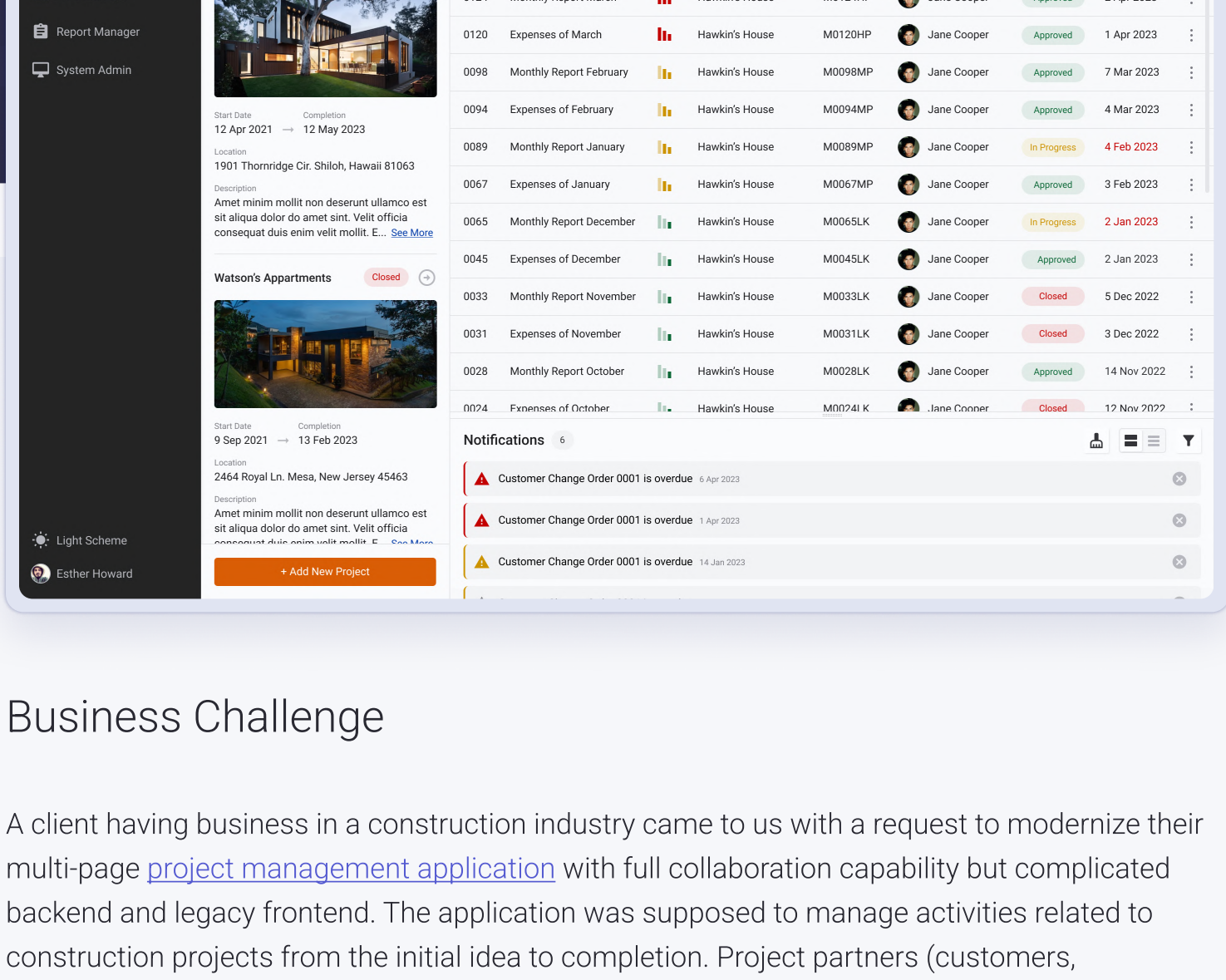


UI/UX Modernization for a Construction Management Web App

A comprehensive web application for a construction company that works with vast amounts of data from different sources. The app provides rich data table configuration, reporting, and analysis functionality.



Business Challenge

A client having business in a construction industry came to us with a request to modernize their multi-page [project management application](#) with full collaboration capability but complicated backend and legacy frontend. The application was supposed to manage activities related to construction projects from the initial idea to completion. Project partners (customers, stakeholders, contractors, subcontractors, etc.) should be able to access the toolset it provides and ensure the right person has the correct information at the right time. The application has been developed to be data-driven, so the content could be generated properly by turning on and off the functionality for different users. The system's specialty had the following:

- ✓ Many collaborative project management features, including different communication methods for project participants;
- ✓ Document flow management (route, approve, review, and update project documents);
- ✓ Document and file management (with the possibility to integrate with cloud storage);
- ✓ Alerts;
- ✓ Project budget management, cash flow management, EVM, and forecasting.

We saw that the application had a great potential, however, there were many challenges that we needed to deal with.

The application has been on the market for more than 20 years, containing several integrated services and tricky dependence and computing logic. During this period, multiple functionalities have been added in an attempt to update the system and meet users' needs based on the feedback they left. It was often done by different software development companies, so the system did receive an impressive set of features. However, all developers used their unique approaches resulting in a **lack of unified UI/UX design** across different system parts. As a result, the app didn't follow the UX trends considered a must-have nowadays.

The flow of the app functionality was stiff, lacking user-friendly UX features but having **troublesome and old controls** that were slowing down the operations of employees. The **icons in the drop-down settings menu were bulky and incoherent**, which was confusing for most of the users. There was **only a white theme** without an ability to switch between white and dark themes or change theme colors to make every user set the app the way they wanted.

When the client came to us and showed their app, we also noticed that **all the forms were static**, and no dynamic form was found. It complicated the user experience even more, considering the fact that about 99% of the application contained tables and forms. Our client also shared the details with us that the buyers of their application wanted to have an ability to configure those forms and tables and adapt them to their business requirements. Despite the common desire for updates and new and convenient features, some of our client's **customers weren't ready for revolutionary changes**. And the client was afraid to lose those loyal partners.

After we gathered all the challenges and requirements, the project goal became apparent to our team. We decided to [modernize the application's UI/UX](#) but make it possible to switch between new and old versions of the frontend and ensure a smooth transition to new features. It was essential to ensure users won't feel like they work with different apps when switching between various pages. Also, we didn't forget that users would like to have complete control over table and form configuration, including the changes in the displayed data, the creation of roles for specific project participants, flexibility in configuring access rights, etc.

Solution

The lack of documentation, which is a usual thing for legacy software developed by multiple teams, was the first challenge to overcome. Our business analyst could only rely on the user guide to **prepare a Scope of Work**. However, in close cooperation with the client, we fully understood all the specifics of the system. And SOW allowed our team to identify all possible app modernization options and consider all the customer requirements.

It is vital to mention that the application had to work with large amounts of data presented in tabular form. Therefore, our most significant task was getting rid of those static forms and **implementing configurable dynamic tables and forms** that can be adjusted according to the users' needs with the fewest restrictions.

The idea of implementing and using dynamic forms consisted of a simple plan: the complete configuration of the form elements (type, template, width of elements, etc.) comes from backend to frontend. Each component had a sequence, showing odd ones in the column on the left side and even ones in the right column.

The dynamic approach **allowed an administrator to have control over the app's appearance right** from the backend. All the styles were saved on the frontend, while all the elements were loaded from the backend. The approach was simple to put into action, because our client's developer knew the system's backend side well and was able to make changes in API. Besides that, their backend was already able to send table configurations on the frontend. To improve the app's UX and entirely **rebuild the front-end with ease**, our development team used [Webix JavaScript UI library](#). It was possible for us to adapt Webix to this functionality and bring our plan to reality.

With the help of Webix, we were able to include, for example, adding and removing columns to a table or fields to a form. In addition to a clearer look, our developers also gave app users easy access to a set of useful functions, such as global search with highlighting of searched fields, complex filtering, multiple sorting, display data configuration, and others. To improve the overall workflow and improve how users operate with documents and other files, we added filter templates. They helped to make the process faster and easier and, if needed, a user is now able to save favorite templates to find documents with just one click.

Bulky icons of the old app did not always make the purpose of certain buttons clear. For a better user experience, **our team has reworked all the old controls** and added text buttons, which are now directly informing users what they are for. Besides that, we added **two versions of the app (white and black)** and the feature to set **custom themes**, choosing the desired colors and functionality.

Lists of users related to a specific project were supplemented with photos, by clicking on which you can now access additional information, such as the role on the project and contact details. We also added the possibility to **set roles to users** and configure their access rights. All changed settings can now be saved as user templates locally in a few clicks. If needed, users can roll back to the default settings received from the server. To ensure a gradual and painless transition to the new user interface, our developers also added the **ability to switch** between its old and familiar version and the new one.

Implemented Components

Taking a closer look at the complex Webix components we used, we want to pay attention to the three key management tools.

Thus, to enable advanced reporting, our developers used the [Webix Report Manager](#) widget. Before the app modernization, reports were available only in the form of tables and can be configured only by administrator. We allowed all users to create and configure reports of any complexity as they want by implementing manager rights as default for all users. As a result, they now have a possibility to set various data sources and define them, create and save filter queries, group, filter, and sort data by different criteria, etc. They are also able to visualize reporting data as configurable tables, diagrams, or heatmaps, and reports could be exported into various formats, including CSV, Excel, and PDF. Among other advantages that the Report Manager gave to our client, we can mention:

- ✓ Easy access via the web browser;
- ✓ Efficient work with large and complex datasets;
- ✓ An extensive set of data operations, including counting, comparing, and giving the average values.

[Webix Document Manager](#) was used to ensure comfortable work with cloud file storage. Its broad customization abilities allowed the development team to use it in two ways. First, as a separate widget for working with files. Second, as a separate block on the document page for attaching files to it. This widget also allowed searching files, viewing, editing, and adding tags, which now helps to organize files and folders. Users can comment on files and share them with their colleagues to improve teamwork. Additionally, Document Manager allows to:

- ✓ Switch between old and new layouts;
- ✓ Make quick on-the-fly changes in files;
- ✓ Play audio and video files;
- ✓ Use the preview panel for quick navigation.

To help admins efficiently manage user roles, our developers used the [Webix User Manager](#) widget. With its help, the administrator is now able to quickly add new users with specific access rights and permissions or adjust rights for existing users. Provided functionality also includes:

- ✓ Tracking all user actions and built-in security audit;
- ✓ Analyzing and summarizing user information;
- ✓ Automated identification and authorization.

It is also important to mention that our client's backend developer added new API end-points to easily use our complex Webix widgets. We explained in detail which API end-points should be added and which URL was requested to get the data for all three widgets. They added them, and we were able to successfully integrate all the updates to the application.

Project in Figures

📅 **6+**
Month
Duration

🕒 **1500+**
Estimated
man-hours

Applied Technologies

Result

XB Software team has implemented a more user-friendly application interface and enhanced user experience. The new UI/UX design guarantees that users won't get lost in many similarly-looking tables, forms, and controls. Webix allowed us to implement a more flexible configuration of data tables without sacrificing application performance. Replacing several separate modules with [complex Webix widgets](#) helped to follow a more straightforward development approach. It resulted in accelerated development time and reduced costs. It will also help simplify the application customization in the future.

Your questions and requests are always welcome!

Visit <https://xbsoftware.com/>
✉ sales@xbsoftware.com
Contact Us →