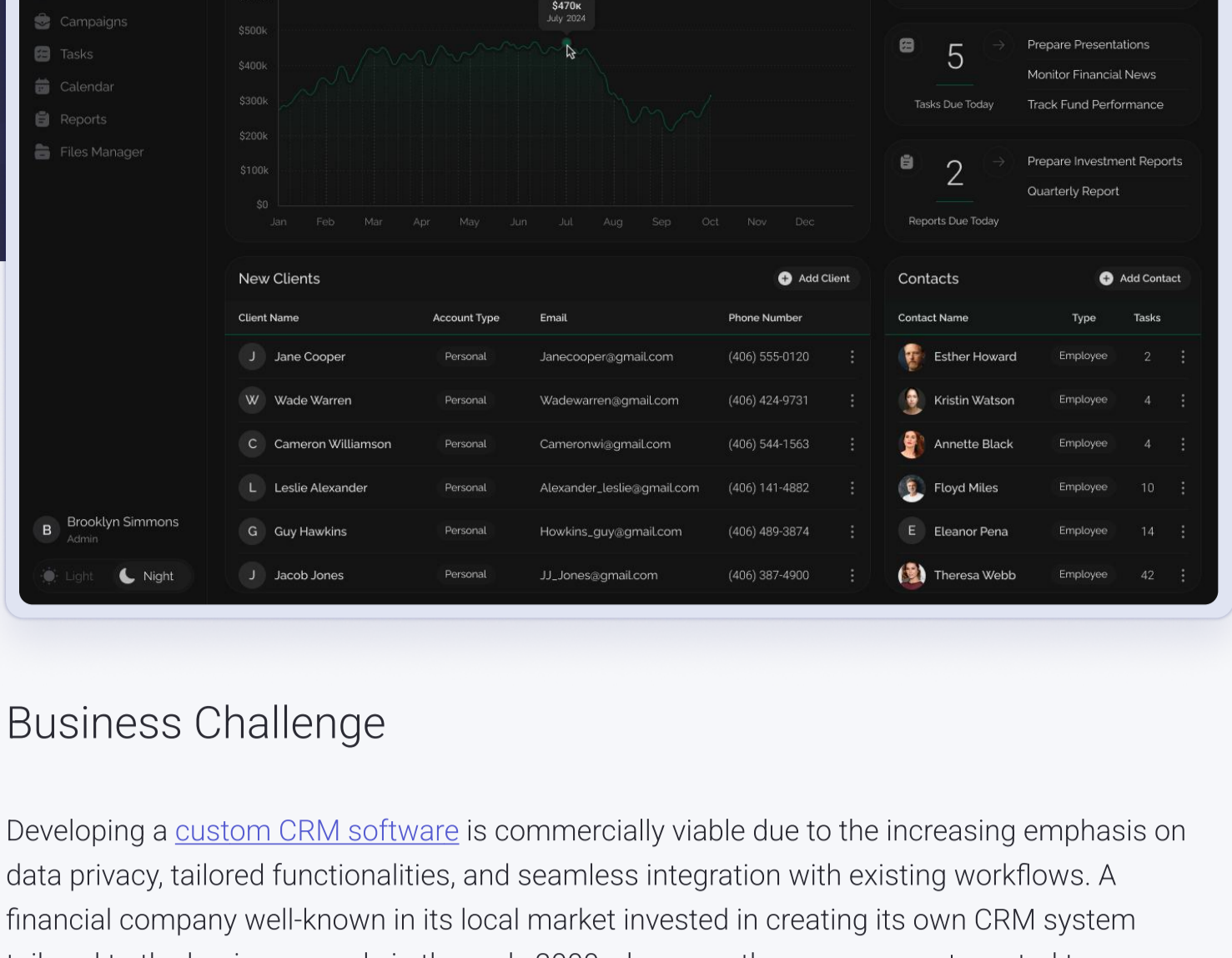


Custom CRM for Financial Services

A custom CRM for a financial company, built for the expansion of existing business logic, with the new features a client lacked previously. The modernized financial CRM system offers better customer relationship management, stronger client loyalty, and improved operational efficiency.



Business Challenge

Developing a [custom CRM software](#) is commercially viable due to the increasing emphasis on data privacy, tailored functionalities, and seamless integration with existing workflows. A financial company well-known in its local market invested in creating its own CRM system tailored to the business needs in the early 2000s, because the management wanted to secure the company's data from sharing it with third-party platforms. Surprisingly, years later, the system was still able to offer useful features and rather flexible settings for financial specialists compared to ready-made but unified Customer Relationship Management (CRM) solutions.

However, the company recognized the need to modernize its outdated financial CRM system, because it was struggling with several issues:

- ✓ **Inefficiencies.** The old system required extensive manual data entry and was prone to errors, leading to wasted time and resources.

- ✓ **Lack of Integration.** The CRM lacked integration with other business tools, complicating workflows and processes.

- ✓ **Limited Analytics.** The system did not provide advanced analytics needed to understand customer behavior and preferences, limiting the company's ability to offer personalized services.

For example, the old system required extensive manual data entry and was prone to errors, leading to wasted time and resources. Also, employees realized that they had to perform many operations manually, which led to low adoption rates and inconsistent usage across the organization. The existing CRM for financial services did not provide the advanced analytics needed to understand customer behavior and preferences, limiting the company's ability to offer personalized services.

These issues hindered the company's ability to deliver personalized services and respond swiftly to customer needs. Therefore, our client decided to rebuild it with modern technologies, expanding existing business logic and adding new features as needed.

Solution

In order to provide the best solution for our client, we evaluated the existing applications' architecture, codebase, and dependencies. During the analysis of our customer's requirements, we explained that when customizing an existing system to suit their needs, we will need to consider the existing limitations of the basic logic. It is possible to face severe restrictions and an inability to fully implement necessary features, or encounter unpredictable errors due to added functionalities.

It was also important to consider the commercialization of the future application, because the decision affects the initial architecture of the application. Defining the payment and distribution model is as detailed and decomposed as the development process itself. There are two main options:

- ✓ Creating it for own company and also for distribution;
- ✓ Developing a brand-new system specifically for distribution.

Depending on the choice, we would have to build a certain system architecture. Initially, the client wanted the first option, therefore, it was the one we followed. In case, for example, the client wanted to be able to distribute the software after we already built it for their own company only, it would be a difficult task, because the system architecture wasn't developed for this change. Our client eventually decided to not use the system for distribution as well but just leave it for their business only. Nevertheless, they are still able to do that at any time, because initially the intent was clearly stated.

As a result, together with the client, we decided to focus on the following key areas:

- ✓ **Integration.** Seamlessly integrate the financial Customer Relationship Management system with other business systems. This would ensure a unified place with all the required data that could be managed easily.

- ✓ **App modernization.** Improve the current features of the client's app and add new relevant features. This would ensure that the CRM system is customized to align with the company's unique workflows.

We created a sequence of steps to address these areas, involving phased implementation to minimize disruption to ongoing operations. Thus, our development team had to alter and rewrite parts of the code to improve the finance CRM application performance, maintainability, and integration capabilities.

Database

The database of the application, which the customer has been using for so many years, was very valuable. But not everything from the legacy application could be useful for a modern one in the same view, that is why the customer should have decided what should be saved fully, partially, or as local extractions.

The process of **data migration** usually takes about 2 or more weeks per one module, so it was up to the client to decide which database content to migrate. For example, there might be a case where names of columns do not match within one table by a mistake of previous developers. There might be a case where SQL injection should work good, but it causes crashes instead or loses some batches of information in the simplest area. This work needed much attention, because analyzing and prioritizing the data required for the future application influences its operation.

Technologies

The technology stack was chosen based on the principle of **keeping what works** and **upgrading what is no longer supported**. The backend, written in PHP, was updated to a modern version, and parts of the code that were not working well were refactored. The database and the whole structure of MySQL injections were similarly updated. For the frontend, various frameworks compatible with PHP were considered, with [ReactJS](#) being chosen for its flexibility. It can be used with various libraries and frameworks, allowing developers to choose the best tools for their specific needs.

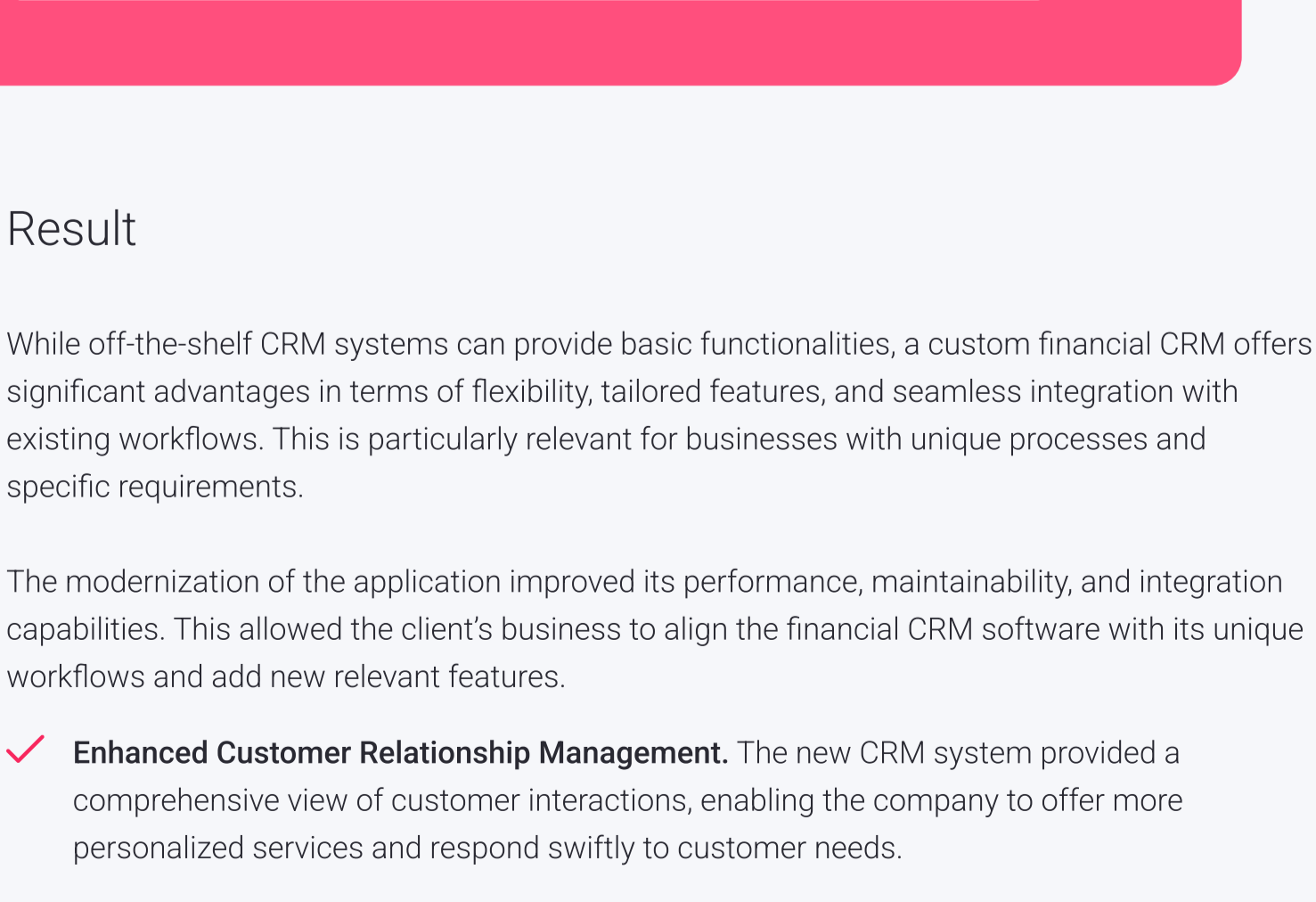
The choice of ReactJS had been made also because of its component-based architecture, which allows for reusable and modular code - and that was a crucial point for developing the application that expects to be scale large.

Main features

The financial services CRM system was integrated with the company's existing financial software, email distribution systems, and customer support tools. This ensured a **unified platform** where all customer data could be accessed and managed efficiently.

Another point to keep the focus on - storing of expanding data. The CRM software is constantly being filled with data, and the amount of data is growing exponentially. Storing this data requires resources, and additional resources are needed to process it. Hosting the database for the application on a **separate cloud storage instance** has increased processing speed. Simple solution for a simple task.

The final point is the volume of information being processed. Since this finance CRM is aimed at handling financial data, every detail is used for analytics as part of an individual user cohort, family, or household. The **output of the results after analysis is performed quickly enough** with the number of records required to be processed.



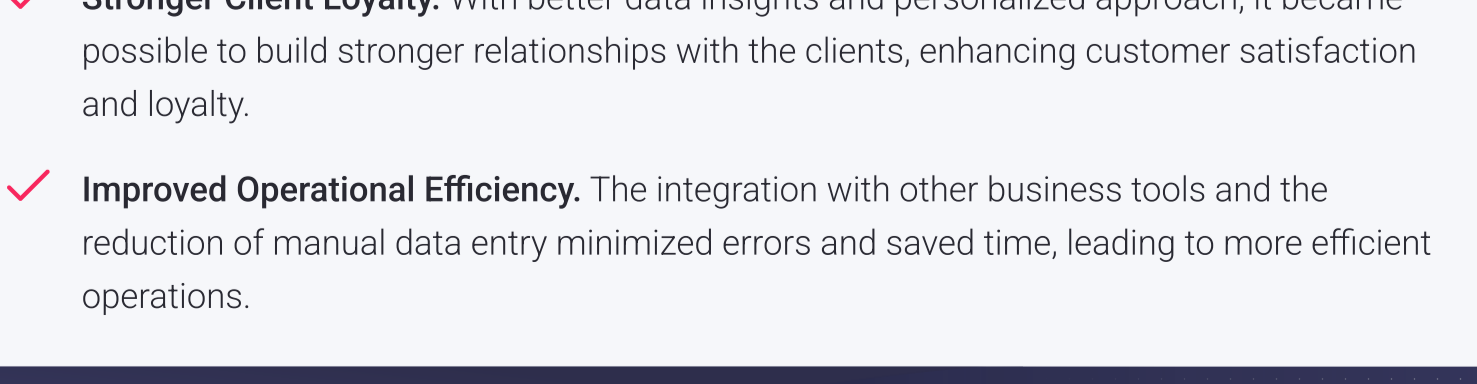
Project in Figures

12+

Month
Duration

9000+

Estimated
man-hours



Applied Technologies

Your questions and requests are always welcome!

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