HENRY SCHEIN:
FINANCIAL PORTAL
FOR GLOBAL HEALTH CARE LEADER

Overview

Henry Schein, a global leader in the health care field, has nearly 20,000 employees, operates in over 20 countries, and has around $10 billion in revenues annually.

Up and Running was tasked with repairing and modernizing "The Portal," the system of record for Henry Schein's Global Treasury Foreign Exchange Balance Sheet and Cash Flow data, a **very important system that processes a large amount of critical financial data.**

Up and Running's work on The Portal is an example of our repairing and improving upon a pre-existing, broken system. (Their business workflow itself was well-designed, just not enabled by technology.) Our efforts left them in far better shape than at the start of our relationship.

Context

The system was brought to us in a state where there were regular issues, and the end users were losing confidence.

Further, prior to our taking over support and extension of the system, the outdated code was failing to inform the end customer of system errors or dropped connections. It provided a very frustrating user experience and it was losing money for Henry Schein and its customers.

Our task was twofold: to stop the hemorrhaging, and to **implement a more modern and stable product and experience for the end users**.

Challenges

After the reliability was addressed for the short-term and long-term, the primary challenge was dealing with a large amount of data in real time to ensure a positive user experience. The more the users need to wait, the more money is being wasted and people lose attention and get upset.

Another challenge, though its implementation went hand in hand with creating a responsive (fast) system, was **designing the system for scale and to gracefully handle future requirements** since it's a very custom application. **Since continual technology improvement and development were important goals of the customer, we made it happen**.

Process

We fixed the immediate problems, implemented best practices, and created communication mechanisms to help the users gain confidence in the system.

UAR focused on **upgrading and improving the application's reliability**, implementing testing policies in the process to improve, sustain, and validate what's in place and what's to come from a system feature perspective.

Through our efforts, we greatly improved the application's notification system and format handling between Henry Schein and its end customers.

Software upgrades and an extensive testing process now ensure that security and reliability issues of all types will be addressed to the extent they want now and forever.

We wrote unit tests to help ensure project integrity going forward, removing the fear of further altering the code in later stages. That is, when any new feature is added, **these tests will guarantee that the rest of the application is working before integrating them**. Also, for any new feature, there shall be additional tests created for that feature so that every aspect of the application is covered.

In order for the application to rapidly recognize and respond to end-client needs, in addition to being able to gracefully handle future requests, we developed a **well-architected and sustainable system** that enables and assesses the quality of customer communications in real time.

Provided with all the necessary data, the application gives an accurate and current financial position by means of **improved forecasts, progress tracking, information gathering, and business growth opportunity discovery**.

To maintain the web services' high level of performance and security, the newest technologies were used and implemented according to best practices.

To enable flexibility for handling future feature requests and to reduce maintenance costs, proper architecture was followed, primarily separation of concerns.

For **an upgrade that was both time-efficient and cost-effective**, we used NodeJS, which was also the language we inherited with the legacy code.

There would have been a large cost to change, but thankfully the language was not at fault for the then-current state of the system.

Solution

NodeJS enabled a great user experience, quickly supplying relevant information to the system's users.

We used **NodeJS** as a cross-platform runtime environment for developing server-side web applications and to improve web service response.

Suitable for full-stack development, NodeJS is a lightweight solution that offers an asynchronous I/O API, complexity of closures, raw speed for this particular purpose, and thinner service calls for which the only thing that moves along the wires is the new data.

NodeJS enabled a great user experience, **quickly supplying relevant information to the system's users**.

The **Express framework**, which provides a thin layer of fundamental web application features, helped in developing **a robust API for easy communication** and data exchange and organizing the web application into an MVC architecture on the server side.

Express is nearly the standard for a NodeJS web application because it's fully customizable and designed for heavy browser applications.

A **jQuery** library, with its simple cross-platform support and reuse of DOM objects, was used to enhance front-end development.

Result

Happy clients and happy end clients, as well as clean data that's easily accessible on a stable platform.

A robust, modern application that tracks customers' Cash Forecast, Account Balances, Bank Compliances, and more.

For the end customers, the result is **improved data accessibility and enhanced input response**, as the notification system helps them to stay in the know regarding the statuses of their transactions and whether submissions are successful, as well as enables them to control those transactions easily.

Further, with an upgraded notification system and newly-implemented submission scheduling module, **our client has better insight into its end customers' submissions**.

Via the notification system, end clients receive information on a daily basis according to Henry Schein's defined schedules.

If needed, they are reminded when and where to submit their reports, and they always know where they are in the process, empowering them to gauge what action is most appropriate to take at that time.

About Up and Running

We partner with clients to solve important technical problems by building, implementing, and servicing complex software solutions for customers of any size in any industry.

Started in 1995 while founder Pete Hanson was a sophomore in high school, Up and Running Software began as a technical services firm and evolved into a developer of custom software solutions. Customer happiness and open communication have been the focus since day one, plain and simple, resulting in long-term relationships with any type of client, from non-profits to startups to the Fortune 10. Thanks to this mindset, Up and Running is fortunate to have served an impressive array of clients, including giants such as General Electric (GE), Sanofi, and Hearst, as well as venture-backed startups at every phase, from formation to growth.

Its over two decades of custom software development, legacy system support, and migrations drive the engines of solutions that thousands of businesses and millions of people rely on daily. That experience has resulted in hands-on depth in most software development stacks, from the command line systems in the back office to ones that gracefully handle any screen size worldwide. A commitment to QA and process improvement drives scalable results, producing long-term software assets with high ROI. Up and Running's work passes FDA & DEA audits, rolls up global treasury data, manages nuclear assets and railroads, delivers vaccines, and enables neuroscientists and researchers.

Contact Up and Running:
solutions@upandrunningsoftware.com
Phone: +1, 888-447-9273
Let us know your preference, and we'll connect you with either a technical architect directly or a non-technical sales representative.