

Remington “Remy” Torres

Software engineer · UI architect · Product designer

I turn complex financial workflows into software that feels clear.

I'm Remington “Remy” Torres, a Bay Area engineer with more than 20 years of experience designing durable interfaces and application architecture for banking and trading environments.

Currently building

Bank Ledger

A local-first financial desktop application, designed and architected from the ground up.

20+ — years in financial technology

26K+ — users served at Wells Fargo

109 — automated Bank Ledger tests

1 — designer and architect behind Bank Ledger

Selected work · 2025—present

Bank Ledger

[Read the case study](#)

An opinionated, local-first desktop tool that brings everyday checking, budgeting, recurring expenses, reminders, and credential storage into one focused workspace.

I am the sole architect and designer of the product's look, feel, interaction model, and system structure. I use OpenAI Codex to help implement my designs. I approve every Codex step and personally read the code for integrity, retaining ownership of every product and architecture decision.

- Designed a modular event system that separates raw interface events, application behavior, rendering, API routes, and storage.
- Architected a local-first Electron application backed by Express and SQLite, with multiple ledgers, backups, restoration, and recoverable deletion.
- Defined security boundaries including sandboxed renderers, per-launch API sessions, contained navigation, validated payloads, and OS-backed credential encryption.
- Created a responsive, themeable ledger experience with budgeting, smart vendor/category workflows, configurable tables, and keyboard-accessible contextual help.
- Built a file-driven localization system and cross-platform release pipeline with automated package verification.

System map · Local-first

Desktop shell — Electron

Internal API — Express

Persistence — SQLite

Modules · Sensors · Behaviors · Validation

Application code: 20K+ lines

Test cases: 109

Languages: 5 included + custom catalogs

Release target: Linux + Windows

JavaScript ES modules · Electron · Node.js · Express · SQLite · HTML & CSS · Tailwind CSS · GitHub Actions

Selected work · Developer tools

Scaffolding

[Read the case study](#)

A reusable starting point for browser applications, built around plain JavaScript and clear responsibilities.

Small modules create the interface. Sensors capture clicks, keystrokes, and other browser events; behaviors decide what happens next. An Express server handles API requests, while storage and other product needs can be added as the application takes shape.

I use this approach instead of React for projects where I want direct control over the page and an explicit path from a user action to the code that handles it. Native JavaScript modules let me work without a JavaScript build step and keep the structure easy for me to inspect and extend.

- Separate interface structure, event handling, and visual styling so each has a clear place to change.
- Register new modules and optional plugins through shared contracts, keeping features organized as the project grows.
- Start with a small foundation and add persistence, authentication, and deployment choices when a product needs them.

As with Bank Ledger, Codex assists with implementation under my direction. I approve every step and read the code for integrity and consistency with the intended architecture.

Interaction flow · Browser-first

User input — Browser events

Shared signals — Sensors

App actions — Behaviors

Factories · Registry · Plugins · Express API

Interface: Native JavaScript modules

Rendering: Direct DOM creation

Styling: Tailwind CSS

Starting storage: Replaceable, in-memory settings

JavaScript ES modules · HTML & CSS · Node.js · Express · Tailwind CSS · Node test runner

Experience

A career built around high-stakes interfaces.

2013–2022 · San Francisco, CA

Senior Software Engineer UI/UX Lead

Wells Fargo · Enterprise Collection and Recovery

- Recruited as the department's first and only UI specialist to modernize a large enterprise collections platform serving more than 26,000 internal users.
- Refactored nearly 1,000 legacy pages, consolidating years of inline CSS and JavaScript into a maintainable front-end architecture.

- Created a global Sass framework and a namespaced JavaScript library that standardized UI development, prevented conflicts, and simplified adoption across teams.
- Established interface standards for one of the bank's largest applications and owned UI delivery across distributed development and testing teams.

2011–2013 · Bay Area, CA

Technology Consultant Freelance

Independent

- Partnered with small businesses to design and deliver responsive websites and approachable, mobile-friendly user experiences.

1999–2011 · Tiburon, CA

Software Engineer

Boosalis Options LP

- Built proprietary trading stations and custom web applications for market makers and financial operations.
- Modernized legacy systems with clearer workflows, custom interfaces, and supporting network solutions.

1989–1999 · San Francisco, CA

Technology Support Trading Floor

First Options of Chicago · Pacific Stock Exchange

- Built, customized, and supported high-performance trading workstations for stock-options market makers during a period of rapid industry change.

Recommendations

Trusted by the people who saw the work up close.

“Remy quickly showed us how the UI should be designed and organized. He set our UI standards and created a smooth, powerful, secure UI that everyone on the team was proud of.”

Patrick Armstrong Former Manager & Project Lead · Wells Fargo

“His code was always well written and clear, with plenty of well-placed comments. Remy was a delight to work with and I wouldn't hesitate to hire him again.”

Rob Mancinelli Former Lead Software Engineer · Wells Fargo

Signed letters of recommendation are available upon request.

Capabilities

Where systems thinking meets interface craft.

Product & UX

Interaction design, responsive interfaces, information architecture, workflow simplification, accessibility

Front-end

JavaScript, HTML5, CSS3, Sass, Tailwind CSS, React, component and event architecture

Application

Node.js, Express, REST/JSON, Electron, SQLite, MongoDB, validation and local-first systems

Delivery

Git/GitHub, automated testing, CI, packaging, technical documentation, cross-team collaboration

Let's make complex things feel simple.

Have a difficult interface or financial workflow?

remcommerce@2k02.net

[\(925\) 639-4656](tel:(925)639-4656)

github.com/remHub

Bay Area, California