

UX design and engineering

I'm a product manager and UX/UI designer who builds working interfaces. I use support conversations, usage data, and hands-on prototyping to connect user needs with the details of implementation.

01 MYTEACHSTONE | SHIPPED

SMS login and account recovery

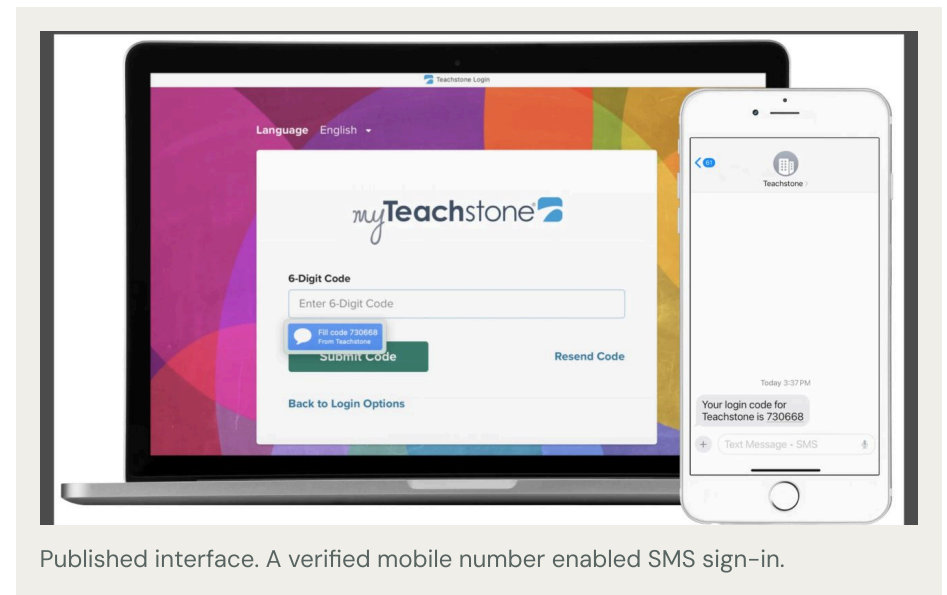
My role: Product Manager, UX/UI design, and prioritization. Delivery with Engineering and Security.

Why people were locked out

Educators were getting locked out after changing jobs: they no longer had the work email tied to their account and could not remember their password. Losing both left them without a practical way back in. Support conversations and case-volume analysis showed a recurring pattern, especially among people less comfortable managing accounts. The result was repeated support requests and educators unable to continue their work.

My contribution

I advocated for prioritizing the fix, designed the login and recovery experience, and led delivery with the team. SMS gave people a route into their existing account when they could no longer rely on work email or remember a password.



About 40%

Fewer login errors

About 80%

SMS adoption among
active users
over two years

Under 4%

SMS-login error rate

From support evidence to recovery

The design question was how to provide another route into an existing account while protecting it from repeated attempts. A verified mobile number became the prerequisite for SMS sign-in.

BEFORE SIGN-IN

Verified mobile number



SIGN IN

Choose SMS login



VERIFY

Enter received code



CONTINUE

Return to the account

Flow reconstructed from the published case; mobile verification happens before SMS sign-in.

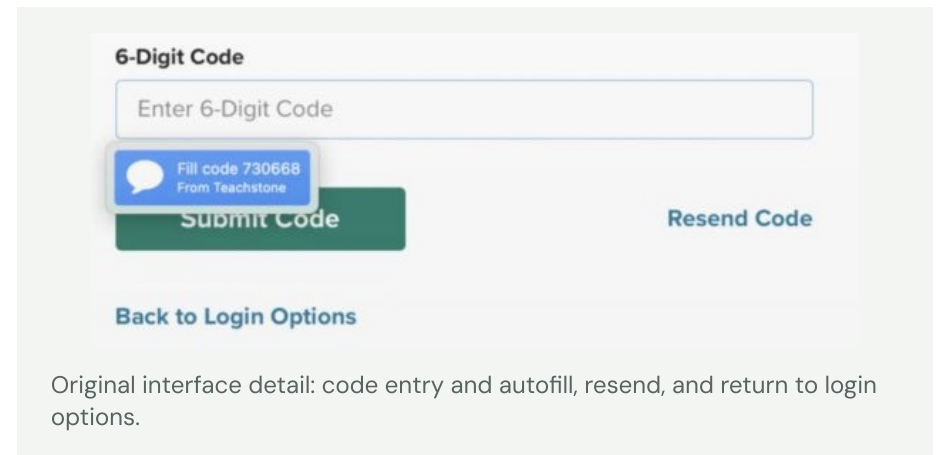
An access problem worth prioritizing

The support evidence connected a routine login failure to an educator's inability to work. I brought that context into prioritization and carried the chosen approach through UX/UI design and delivery. The team's implementation addressed the recovery path and the controls around it.

Recovery at the point of friction

SMS autofill reduces manual entry. Resend Code and Back to Login Options give people visible routes forward when a message does not arrive or they need another method. These controls keep recovery close to the task.

This remains one of my favorite projects because the result was so practical: educators could get into the platform and continue their work.



Helpful errors with safeguards

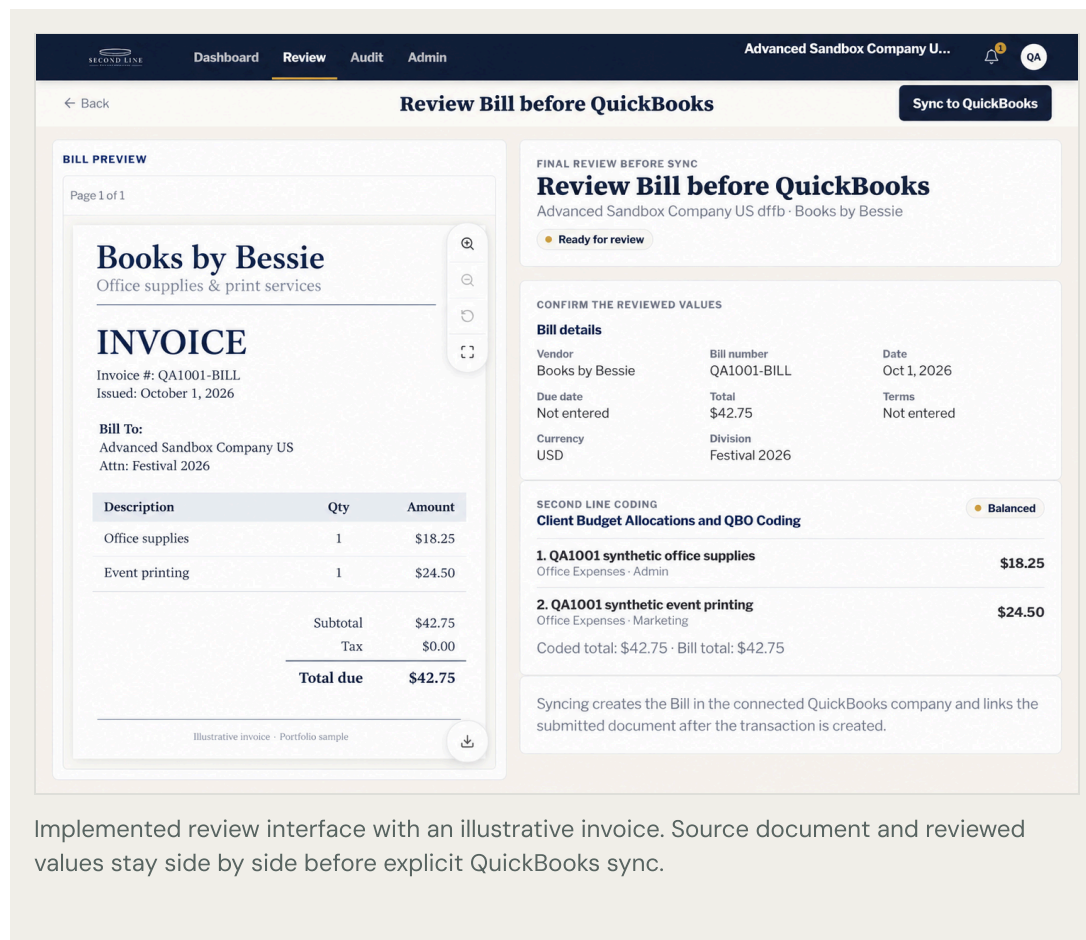
Attempt throttling let us show more helpful error messages while controlling repeated attempts. I worked with Engineering and Security to connect the interaction design with account protection, then followed login errors and SMS adoption after release.

[Read the original project story ↗](#)

Accounting intake and review

My role: Product definition, UX/UI, and Next.js, React, and TypeScript implementation.

An accounting firm needed clients to submit bills and supporting information while staff retained control over review and QuickBooks entry. I worked through stakeholder questions and sandbox feedback, then built the forms, permissions, and submission states.



Implemented review interface with an illustrative invoice. Source document and reviewed values stay side by side before explicit QuickBooks sync.

A clear handoff

Client language such as Project/Event maps to the accounting fields behind it. Reviewers compare the document with structured details before confirming the record. On narrow screens, Document and Details views keep both parts available.

Field errors without interruption

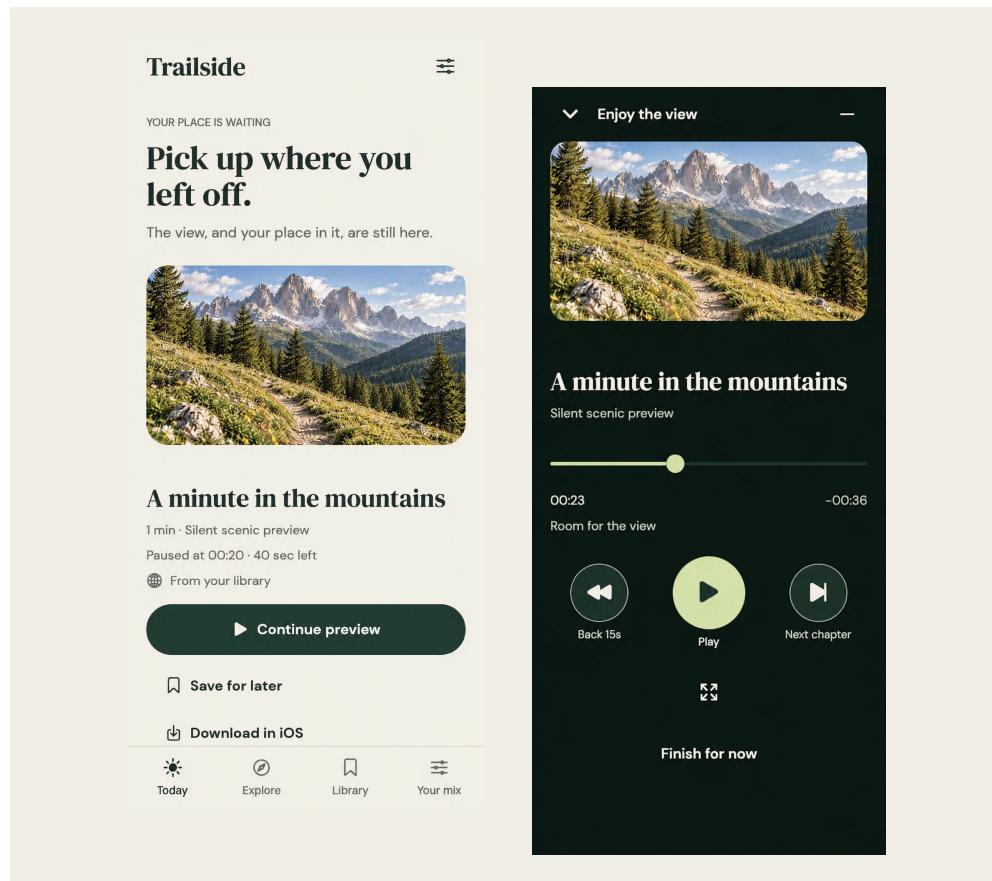
Next, Review, or Submit reveals and focuses the first invalid field. Errors stay beside the fields and entered values survive failure. Background saves leave focus and scrolling alone. Shared feedback components distinguish validation errors from general save failures.

The workflow is implemented and has desktop and mobile QA evidence. That work reinforced a design priority for me: preserve people's work and make the next useful action clear.

Scenic media for indoor exercise

My role: Product concept, UX/UI, React Native implementation, and browser review.

Trailside explores scenery and company during treadmill or bike sessions. I organized the app around a prepared outing, with warm discovery screens and a quiet player for use while moving.



Today prioritizes resume; the player gives transport controls room. Implemented web preview with generated scenic artwork and silent audio.

Availability people can trust

Saved, Downloads, and In progress represent different intentions. A bookmark does not promise offline availability; download readiness requires verified media. Finishing early preserves a place to resume, while actual playback completion marks the outing finished.

A design system carried into code

Shared components and tokens support accessible labels, large controls, landscape layouts, and loading and retry states. Setup is skippable, and unfinished outings take priority over new recommendations. Browser walkthroughs exercised playback, seeking, resume, and recovery. Native iPhone acceptance remains pending.

[More selected projects ↗](#)