AMRITA

SENIOR UX DESIGNER

UX PORTFOLIO

mayuri.amrita@gmail.com

https://www.linkedin.com/in/amritamayuri/

Hi,

I am a senior UX designer with 10+ years of experience, including 7+ in UX. I design enterprise platforms with a focus on usability, collaboration, and meaningful impact.

I am passionate about simplifying complex systems and thrive in fast-moving, cross-functional environments where design can make a real difference.

I am always curious, always learning, and eager to apply my skills to new domains and challenges. I am currently exploring how AI can shape user experiences and help design responsibly for intelligent systems.

Let’s create intuitive, scalable solutions - grounded in empathy, driven by clarity.

2018

Turvo | Senior UX Designer
Hyderabad, India | Feb 2018 - Present

2015

Greysprings | Graphic designer
Hyderabad, India | Jan 2015 - Feb 2018

2014

Foley designs | Internship
Ahmedabad, Gujarat | 2010 - 2014

2010

National Institute of Design (NID) | B.des in Graphic Design
Ahmedabad, Gujarat | 2010 - 2014

2009

Jawahar Navodaya Vidyalaya | School education (Class 6–12)
Chandauli, Uttar Pradesh | 2002 - 2009

CORE UX COMPETENCIES

- UX Strategy & Ownership
- Interaction Design
- Enterprise-Grade Platform Design
- Design Systems & Governance
- Prototyping & Usability Testing
- User Research Integration
- Systems Thinking
- UX Testing & UAT
- Global Collaboration
- Risk- and Compliance-Aware UX
- Self-Serve Experience Design

SOFT SKILLS

- Communication & Presentation
- Collaboration & Teamwork
- Facilitation
- Curiosity & Proactiveness
- Accountability & Confidence
- Compassion & Empathy
- Feedback Handling
- Professionalism
- Adaptability
- Attention to Detail
- Business Awareness
- Stakeholder Alignment

DESIGN SKILLS

- Wireframing
- Prototyping
- User Testing
- Design Thinking
- UX Writing & Microcopy
- Information Architecture
- Usability Research

TOOLS

- Figma
- Illustrator
- Jira
- Confluence
- Productboard
- Slack
- Figma Slides / Keynote

CASE STUDIES

CASE STUDY 1

SMART IMPORT

Redesigned Turvo's data import experience while adding powerful new flows—Create, Update, and Dynamic Assist—to simplify complex onboarding, eliminate template dependencies, and deliver a faster, self-serve solution.

CASE STUDY 2

CUSTOMIZABLE WEBHOOKS

Enhanced Turvo's existing webhook system by designing a flexible customization experience that lets customers define triggers and payloads independently, reducing support dependency and improving integration speed.

CASE STUDY 3

EMAIL TEMPLATES

Designed a brand-new self-service email template system that enables tenants to create, personalize, and manage templates with full visibility—strengthening branding and reducing support tickets.

CASE STUDY 1

SMART IMPORT

Smart Import simplifies data uploads with Create, Update, and guided flows—reducing errors, support, and customers' onboarding time in Turvo.

Admin | Self - serve UX | Onboarding | Data ingestion | Error reduction | UX for complex systems | Scalable design

Role: UX lead

CASE STUDY 1 - SMART IMPORT

OVERVIEW

Turvo, a Transportation Management System (TMS), required new customers to upload large data sets through 70+ rigid files using the Turvo Data Importer (TDI). The process was complex, error-prone, and heavily reliant on support, slowing onboarding and delaying revenue.

Smart Import was designed to simplify this by introducing three key data onboarding and management flows.

Create flow: This flow allows users to import new records into Turvo for the first time. It is typically used during the initial onboarding phase when large volumes of customer data need to be imported into the system.

Update flow: Once data is already in Turvo, the Update Flow helps users modify or update existing records via file upload.

Dynamic Assist: It is a guided, question-based experience designed for new users. It helps users understand entity dependencies and upload prerequisites.

INSIGHTS

Through conversations with users and internal stakeholders, we identified several recurring pain points that shaped the core direction of the redesign:

Faster and Simpler Data Upload
Fewer Files, Less ID Mapping
No More Manual Lookup Code Maintenance
Simplified ID Management
Fewer Template Change Disruptions

“I just want to upload my data quickly, without worrying about how optional files are structured.”

“There are 3,000+ lookup codes. It takes weeks just to get them in the right place.”

“We hold off on template updates because it causes too much disruption for users. We need a way to evolve without breaking things.”

RESEARCH AND DISCOVERY

To uncover key challenges with the Turvo Data Importer (TDI), we conducted discovery across multiple touchpoints:

We **interviewed existing customers** to understand pain points during onboarding and ongoing data uploads.

Our onboarding team shared **internal insights** on where users typically struggled and where Turvo had to invest the most support time.

We grouped and **analyzed user stories and JTBD** across personas, including admin users, new customers, and internal stakeholders.

These insights helped us clearly define the core problems, reduce friction points, and ensure the redesigned experience aligned with real-world needs and expectations.

JTBD

When onboarding a new customer into Turvo,
I want to upload my data without worrying about complex template rules,
so I can get customers onboarded faster and with less effort.

When reviewing my data before ingestion,
I want to preview and fix errors or missing fields,
so I can ensure all records are ingested correctly without system errors.

When migrating away from the old TDI templates,
I want to introduce changes without disrupting customers,
so I can keep TDI up-to-date while minimizing rework and support costs.

When uploading entity data,
I want to upload only one file per entity that includes both required and optional fields,
so I can avoid splitting data across multiple files and reduce ID mapping complexity.

When starting with Turvo,
I want to add my company's data quickly and easily,
so I can start using the platform without delay.

When managing and updating records in Turvo,
I want to avoid manually tracking and maintaining internal TDI IDs,
so I can easily update data without dedicating time or relying on how the data was originally ingested.

When working with TDI,
I want to get guidance throughout the process,
so I can smoothly upload data without confusion or support dependency.

When preparing data for import,
I want to use the files I already have without converting them into Turvo's format,
so I can save time and avoid rework.

DESIGN PROCESS

I collaborated with the product manager right from the discovery phase, actively contributing to research and early problem framing. Once we had defined the core challenges, I began translating insights into actionable design work.

Journey Mapping & Exploration · Mapped user journeys, wrote user stories, and aligned with JTBD to identify friction and prioritize key flows.

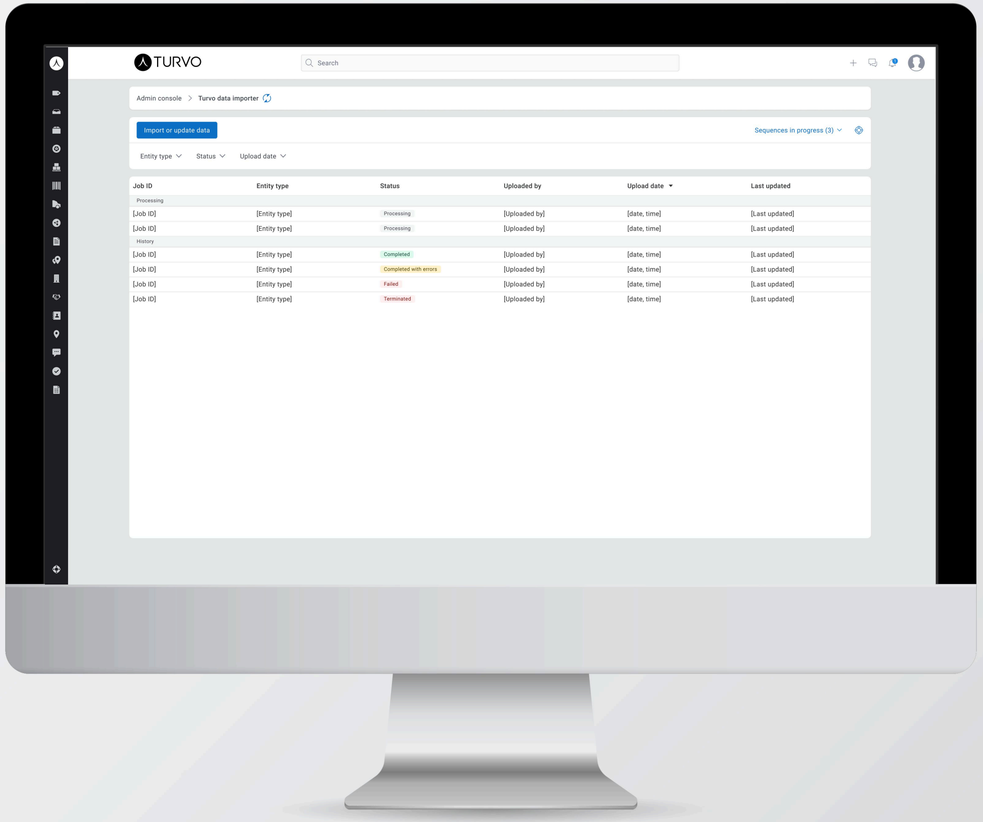
Wireframes & Prototyping · Created low-fi wireframes and interactive prototypes to explore solutions and gather early feedback.

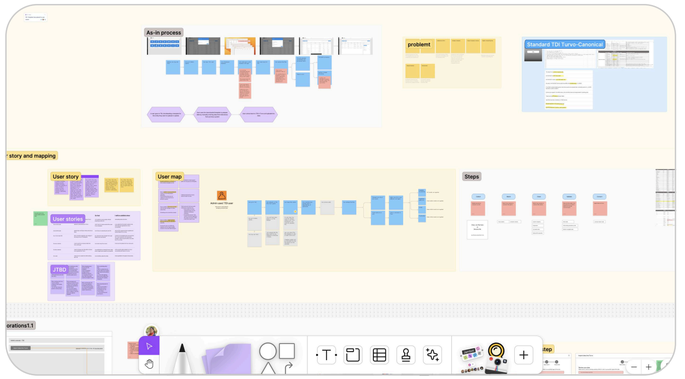
Feasibility & Iteration with Engineering · Worked closely with architects and engineers to ensure feasibility and iterated based on technical and stakeholder input.

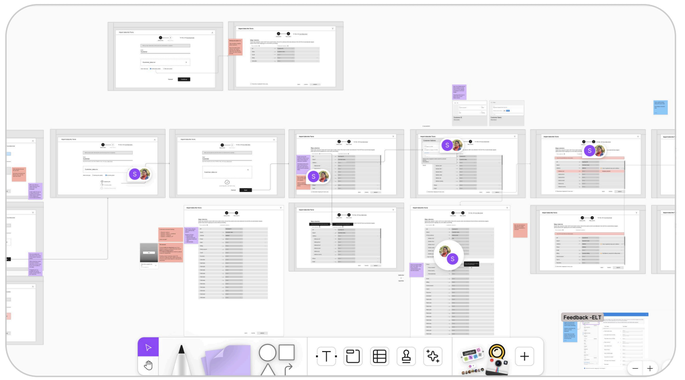
Stakeholder & SME Reviews · Collaborated with subject matter experts to validate use cases across all impacted entities and flows.

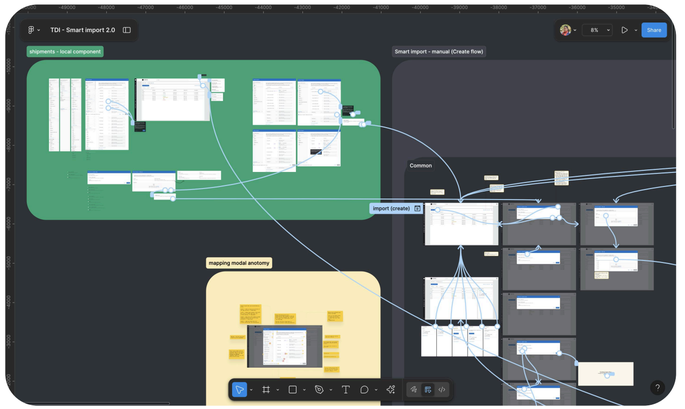
Support & testing · Supported developers throughout implementation, resolving issues and refining designs. Partnered with QA to validate functionality through UAT.

CASE STUDY 1 - SMART IMPORT







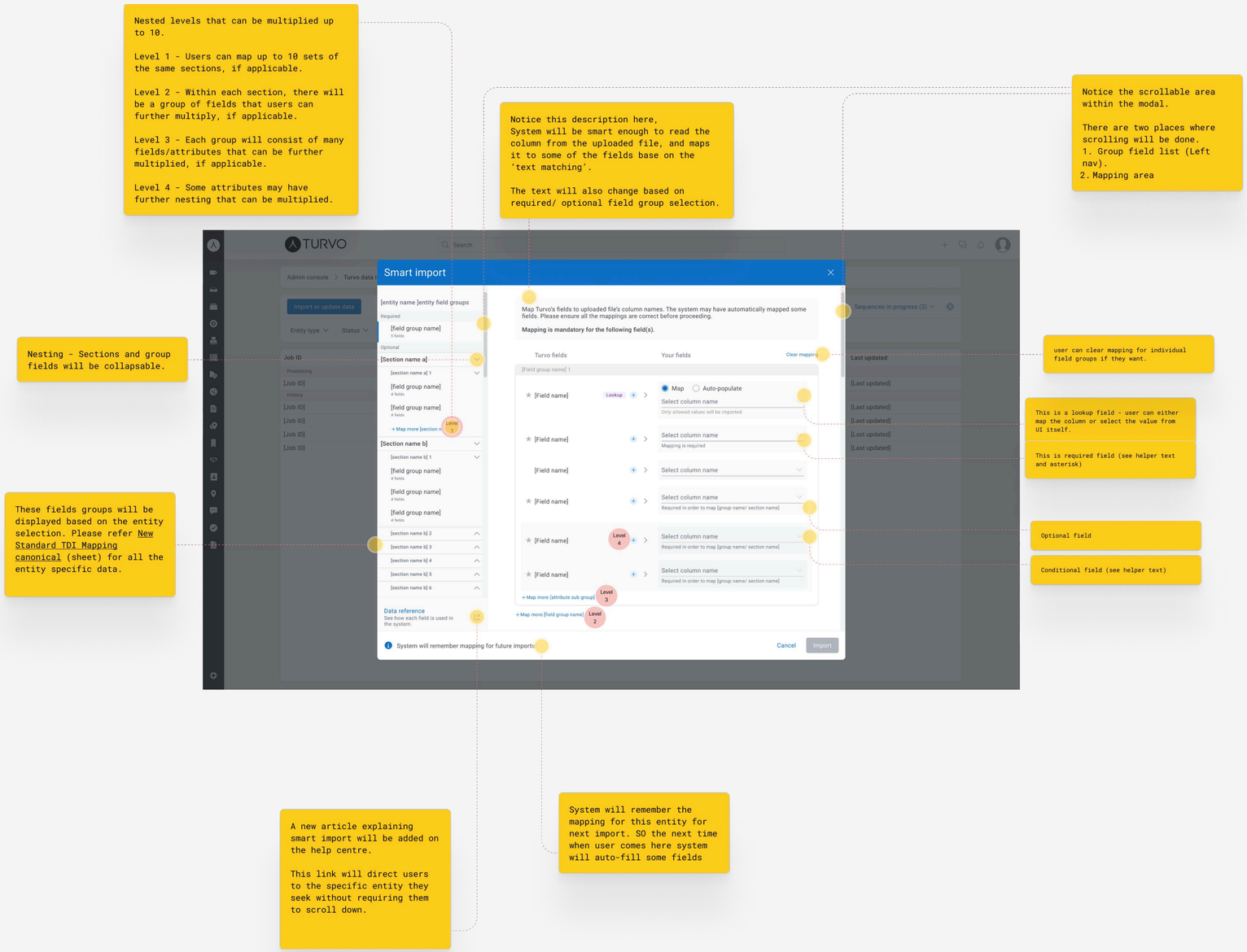


Ideation, wireframing, prototyping and hi-fi mockups

CASE STUDY 1 - SMART IMPORT

I stayed closely involved through development and UAT to ensure the design intent was preserved and adapted to evolving requirements.

I established Figma as source of truth by annotating mockups & providing walkthroughs.
I helped developers understand the feature logic and not just the screens, so we could build the right experience, not just the right layout.



ACCOMPLISHMENTS

✅ **Eliminated rigid template** dependency, accelerating onboarding and reducing manual prep.

✅ Shifted the majority of ID-mapping requests away from support by **streamlining the upload process**.

✅ **Reduced data bottlenecks in shipment imports**, increasing efficiency across workflows.

✅ Built a fully self-serve import experience, **improving user autonomy and reducing friction**.

✅ **Saved weeks** of prep time by removing 3,000+ manual code lookups.

✅ **Improved overall data quality and reliability** through a clearer, guided process.

REFLECTION & LEARNINGS

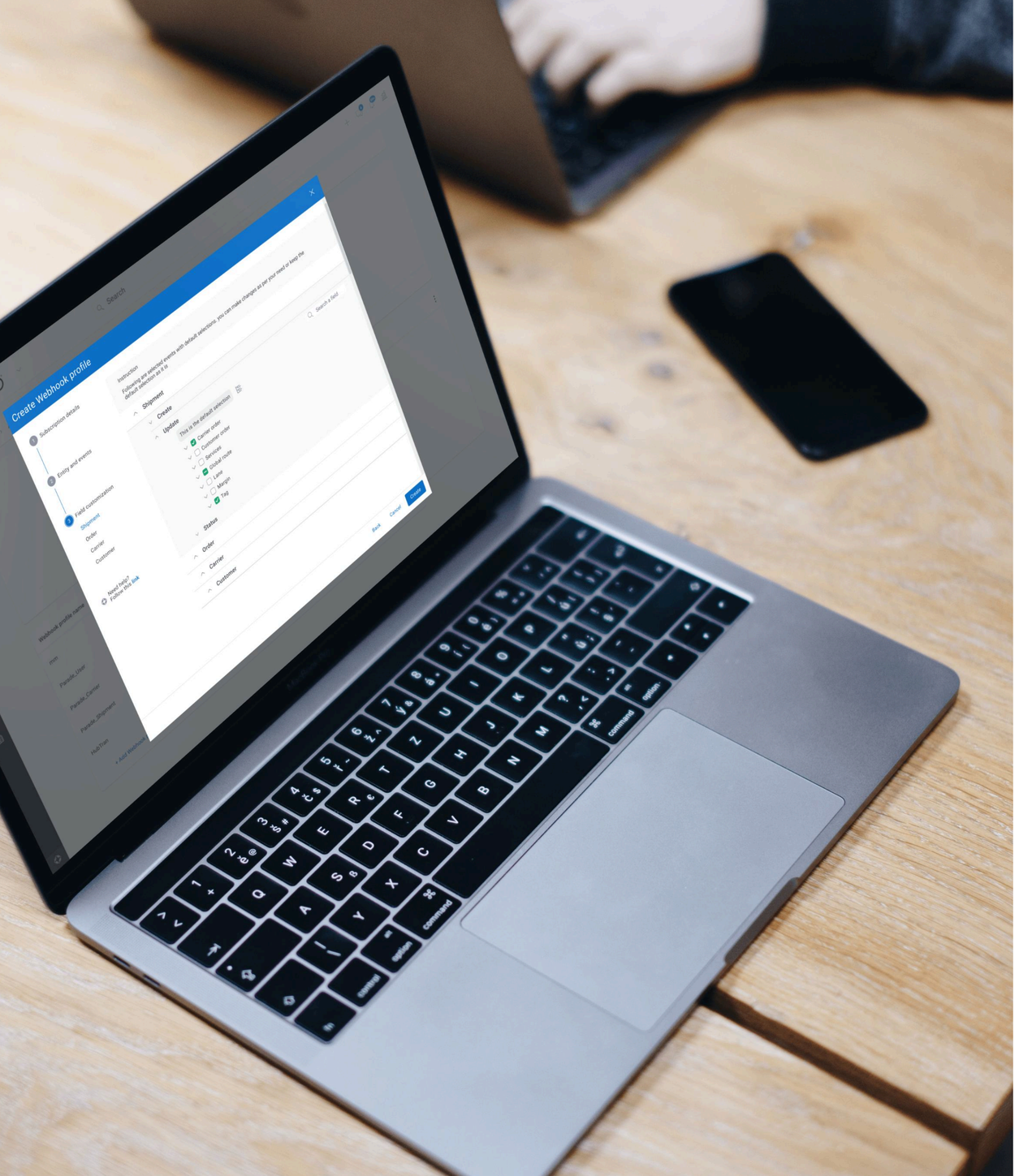
Leading the UX for the Smart Import project strengthened my ability to align diverse teams around a shared vision and drive progress through complex challenges. Daily collaboration with stakeholders enhanced my leadership, sharpened my problem-solving, and enabled me to influence outcomes beyond design, ensuring the final product delivered measurable impact for users and the business. It taught me the value of deeply understanding data-heavy systems, designing for edge cases, and staying hands-on through implementation.

Key Learnings:

Collaboration is critical: Working closely with PMs, developers, architects, and QA helped uncover constraints early and ensured the final solution was both usable and buildable.

Designing for complexity requires clarity: Mapping journeys and JTBD helped break down a complex system into digestible parts and prioritize the right problems.

Support doesn't end at design: Staying involved through development and UAT allowed me to ensure the design intent was preserved and adapt quickly to changes.



CASE STUDY 2

CUSTOMIZABLE WEBHOOKS

Customizable Webhooks let users define real-time triggers and payloads to send data to external systems, enabling flexible, event-driven integrations.

Webhooks | Self - serve UX | Customization | Scalable design

Role: UX lead

OVERVIEW

Turvo's webhook system was rigid and required constant support intervention for small customization requests, creating versioning headaches and unnecessary system load.

We developed a self-serve customizable webhook framework that gave users control over triggers, payloads, and entity types.

This reduced support burden, improved scalability, and enhanced the customer experience by allowing faster integrations and clearer data flow.

INSIGHTS

The existing webhook system had several limitations:

High volume of support requests: Customers frequently requested adding or removing fields in webhook data, resulting in an overwhelming number of support tickets.

Difficult version management: Maintaining multiple webhook versions for different customer needs created complexity and wasn't scalable.

Non-scalable custom requirements: Some customers required very specific changes, which were too resource-intensive to manage individually.

Unnecessary system load: Webhooks were often triggered even when no actual data changes occurred, adding avoidable strain on the system.

These issues led to delays, support overhead, and infrastructure waste, impacting both customer satisfaction and internal efficiency.

RESEARCH AND DISCOVERY

We explored:

- **Support tickets** to identify the most frequent webhook customization requests
- **Patterns across customers** with similar needs
- **Developer feedback** on maintenance issues
- **Backend triggers** to understand when and why unnecessary webhook calls were being made

We validated that a self-serve webhook customization could solve these issues without requiring multiple versioned endpoints.

DESIGN GOALS

Enable self-serve customization: Allow users to define and adjust webhook payloads without relying on support.

Minimize support and versioning overhead: Reduce dependency on support teams and simplify version management.

Enhance control and transparency: Provide clear field-level visibility and granular control over webhook data.

Unify workflows: Support multiple entity types within a single, consistent flow.

Seamless integration with existing experience: Extend the existing webhook feature with customization options that blend naturally into current workflows, without new learning curves or navigation changes.

DESIGN PROCESS

Existing vs. Proposed Flow: We started by analyzing the current webhook flow and identified areas to improve user control and visibility.

Wireframes & Lo-fi Exploration: Designed early layout and structure to support custom field selection and multi-entity configurations.

Feasibility: Collaborated with PM, devs, and architects to define limitations and plan MVP scope.

High-Fidelity Mockups & Prototype: Built a click-through prototype with detailed field hierarchy, instructions, and interaction flows.

User Testing: Conducted usability tests with actual customers to gather feedback and iterate.

Implementation Support: Annotated and organized final designs and stayed closely aligned with dev/QA during build and QA.

KEY FEATURES

Step-based configuration flow

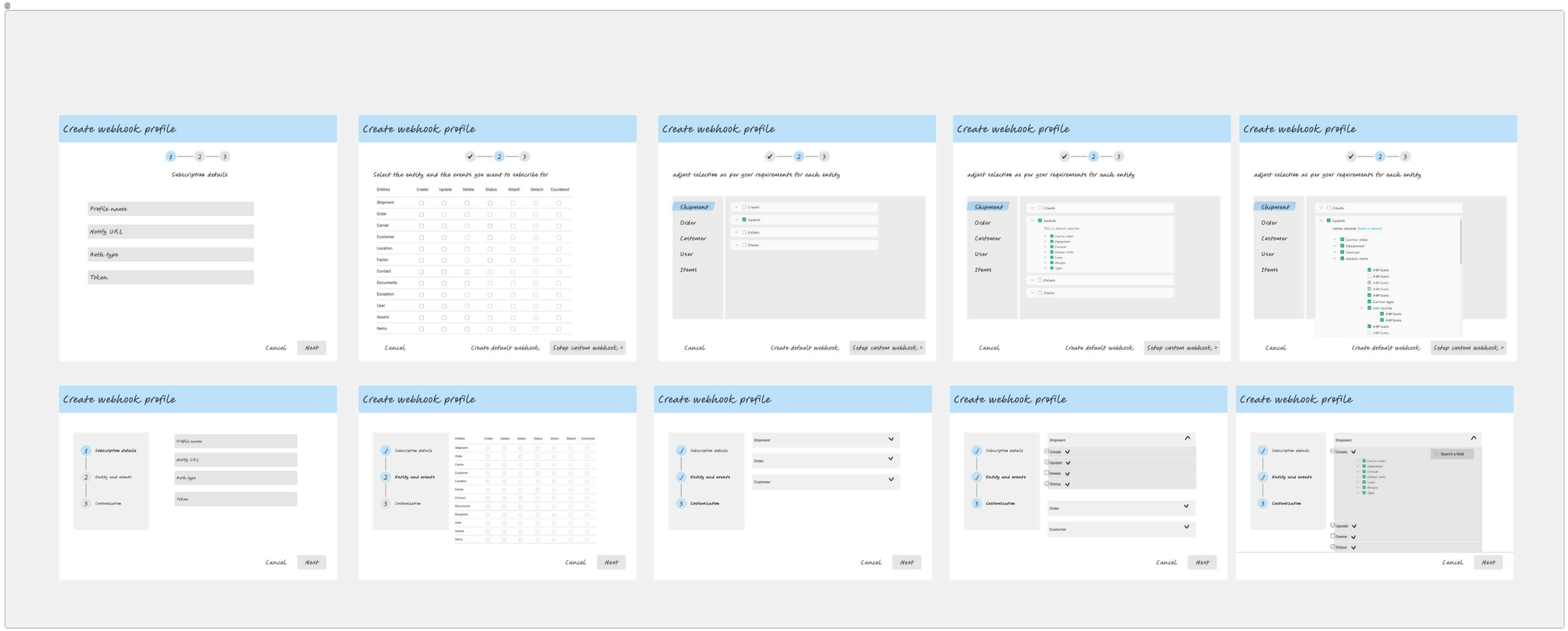
Custom field selection per event/entity

Improved hierarchy and labeling

Clear instructions to guide users

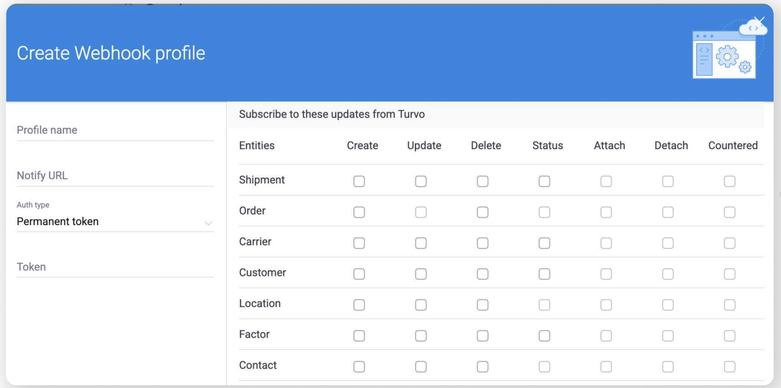
Multi-entity customization in a single experience

CASE STUDY 2 -CUSTOMIZABLE WEBHOOKS

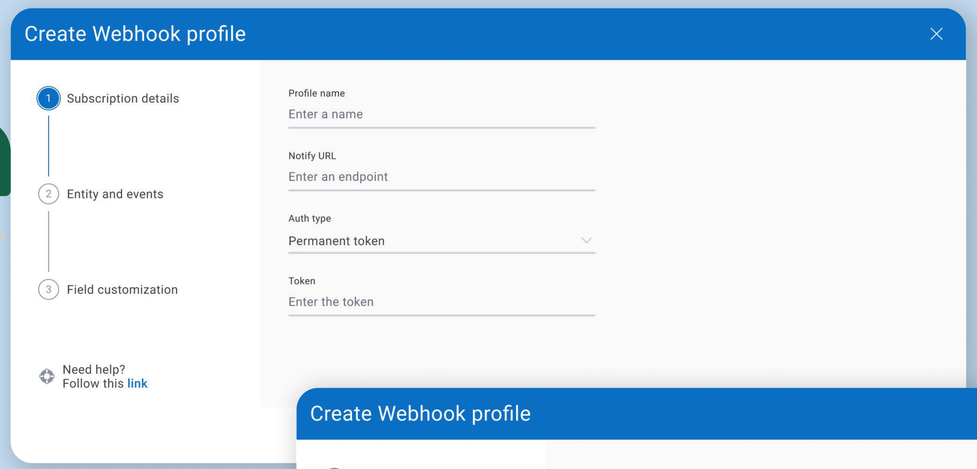


Wireframes & Lo-fi Exploration:

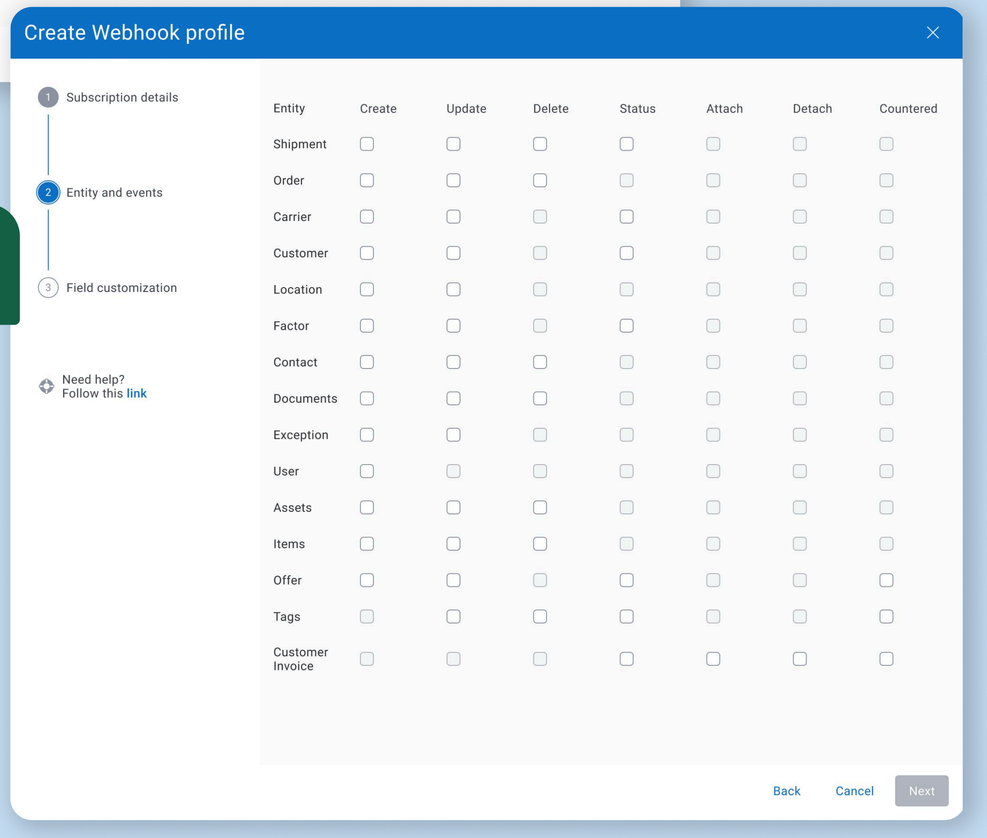
CASE STUDY 2 -CUSTOMIZABLE WEBHOOKS



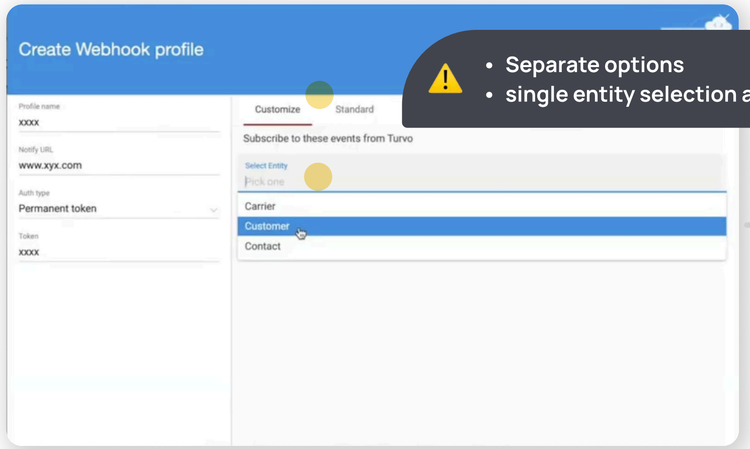
Steps based approach



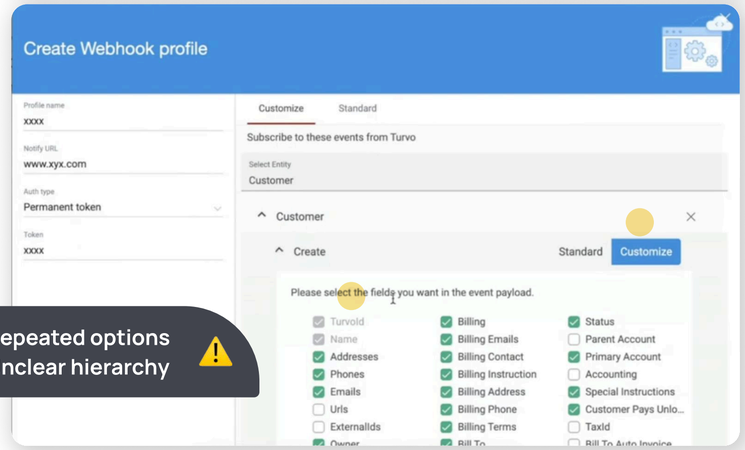
Integrated existing flow with new feature



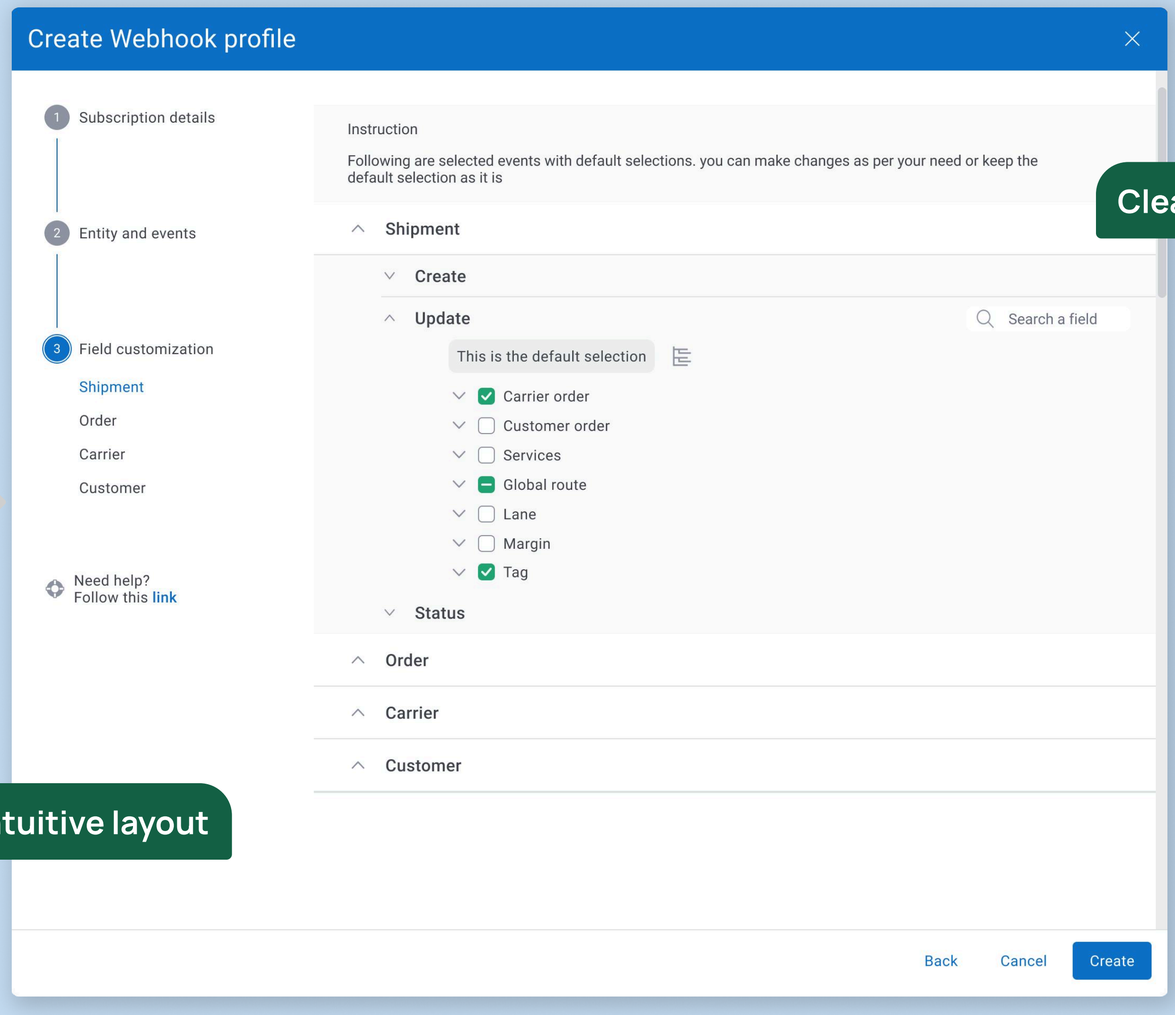
CASE STUDY 2 -CUSTOMIZABLE WEBHOOKS



- Separate options
- single entity selection at a time



- Repeated options
- Unclear hierarchy



Clear instructions

Intuitive layout

ACCOMPLISHMENTS

✅ Shifted the majority of webhook customization requests **from support to a self-service flow**.

✅ Gave users **complete control over webhook** payloads and triggers, improving accuracy and speed of integrations.

✅ **Reduced unnecessary webhook traffic**, easing infrastructure load.

✅ **Simplified version management** by replacing multiple versions with a single flexible framework.

✅ **Increased customer satisfaction** by providing faster turnaround on integration needs.

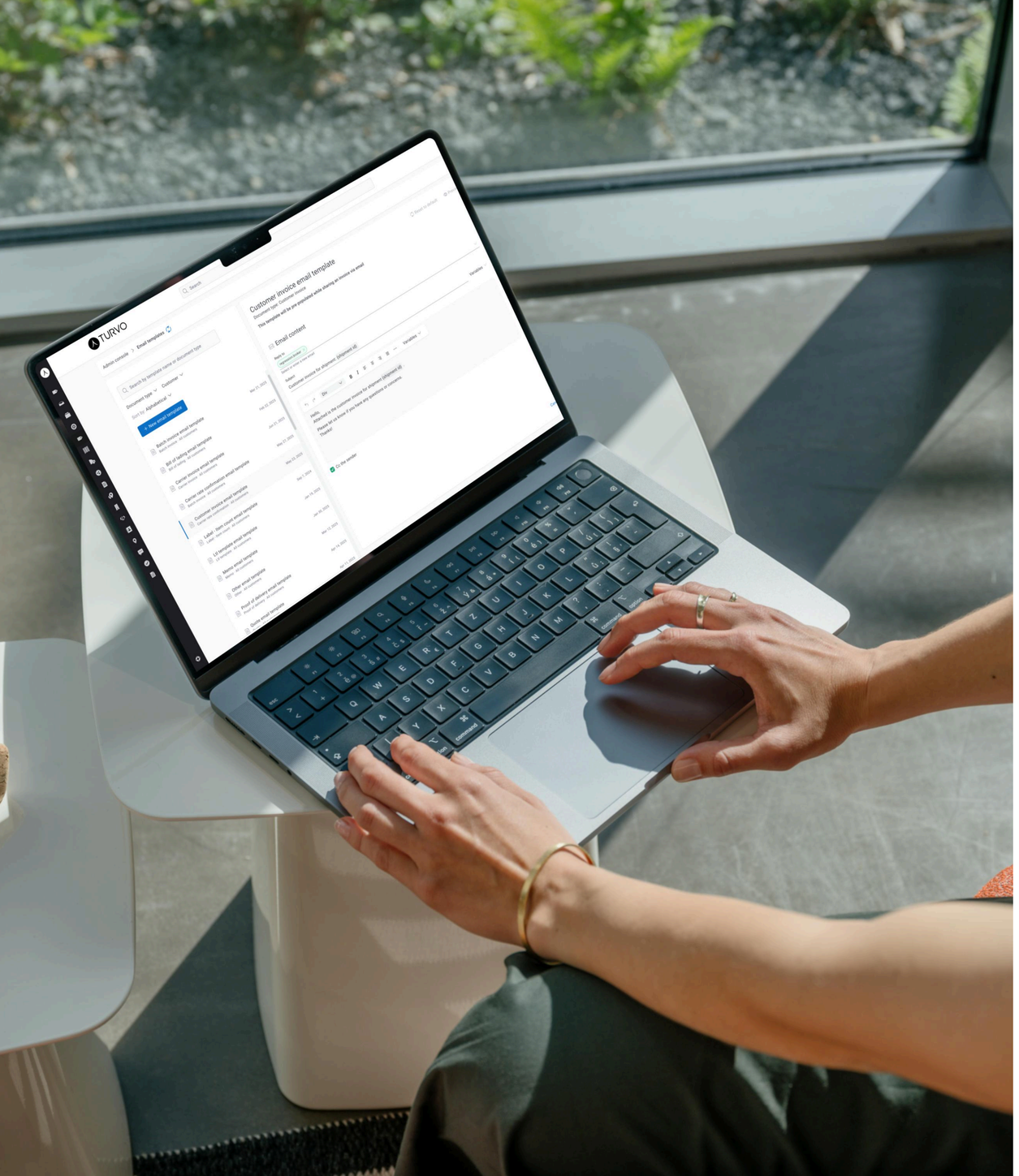
REFLECTION & LEARNINGS

Small changes can have big impact: Even minor improvements in internal tools can create significant downstream effects on scalability and support.

Early cross-functional collaboration is critical: Partnering with architects early helped balance feasibility with usability and avoid costly redesigns later.

User testing builds confidence: Testing not only uncovered the need for clearer hierarchy and guidance, but also ensured we built something users would actually adopt.

Empowering users empowers the company: By designing solutions that give users more control and independence, we ultimately strengthen the company's scalability and success as well.



CASE STUDY 3

EMAIL TEMPLATES

Emailing documents in Turvo is one of the most frequently used features, especially for Shipments. However, customization was limited and often required support. We made the system more robust by enabling customers to create their own templates based on document type, customer, or mode.

Document email | Self - serve UX | Customization | Template management | Personalized communication | Customer empowerment

Role: UX lead

OVERVIEW

Emailing shipment documents is one of the most widely used features in Turvo, yet users had little control over template content or branding, often needing support even for minor changes.

We redesigned the email communication system to allow tenants to build, personalize, and manage multiple templates per document type, with preview and routing controls built in. The new experience improved communication accuracy, strengthened branding, and reduced support dependency for one of Turvo's highest-volume features.

THE PROBLEM

No Self-Service Control
Customers couldn't modify email content or design on their own. Even 'small text updates' required Turvo Support, leading to delays and reduced flexibility.

Limited Personalization & Routing
Users lacked the ability to add CC/BCC, define team-based reply-to addresses, or dynamically insert shipment-specific data into emails.

Static Templates & Generic Emails
Rigid templates produced slow turnaround, frequent manual workarounds, and off-brand messaging that didn't reflect tenant identity.

Painful for Large Tenants
Enterprise customers needed more control, speed, and consistency in their communications, something the old system couldn't provide.

Mismatch Between Preview and Actual Email
Email previews in Turvo often didn't match the received email due to hidden backend templates, causing confusion and inconsistency.

DESIGN GOALS

Enable self-service: Allow users to create email templates without relying on support.

Support advanced personalization: Empower users to personalize email content using data variables.

Flexible communication: Provide configuration options for Reply-To, CC, and BCC fields.

Support multiple templates: Create a framework where users can build multiple templates for each document type based on customer or mode.

Accurate previewing: Provide a preview option so users know exactly what the final email will look like.

Search, sort, and filter: Allow users to quickly find the templates they need.

Consistent platform experience: Leverage the design system to ensure consistency across the platform.

DESIGN PROCESS

Problem Framing: Partnered with PM and Support to analyze tickets and customer feedback to identify core issues.

Journey Mapping: Mapped the full user journey for email template creation and editing to uncover pain points.

Wireframes & Prototypes: Designed and tested wireframes and prototypes to explore solutions and collect early feedback.

Feasibility Checks: Worked with architects and developers to confirm backend support for data variables, multiple templates, and routing rules.

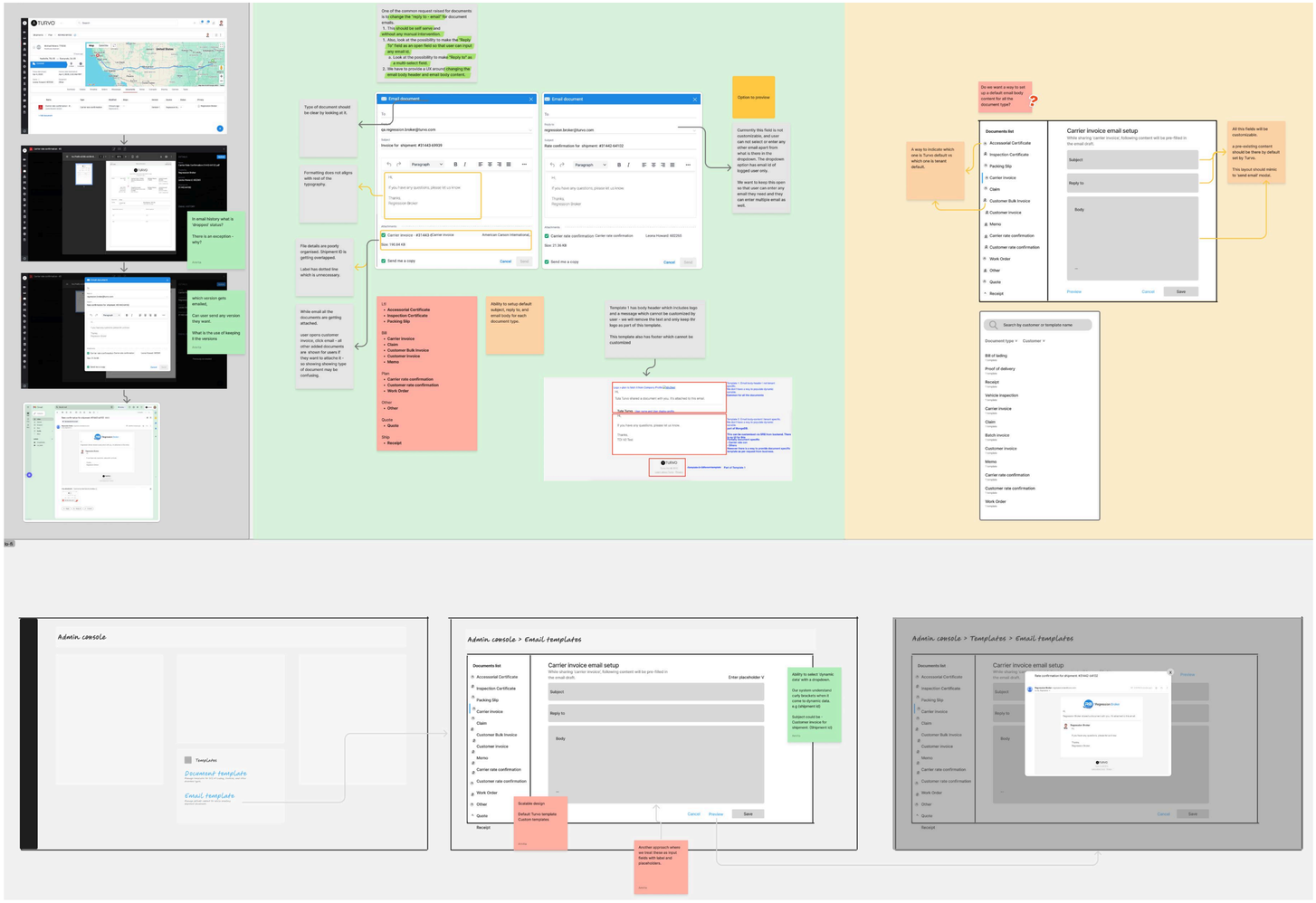
Testing & Feedback: Ran usability tests with internal teams to validate ease of use and functionality.

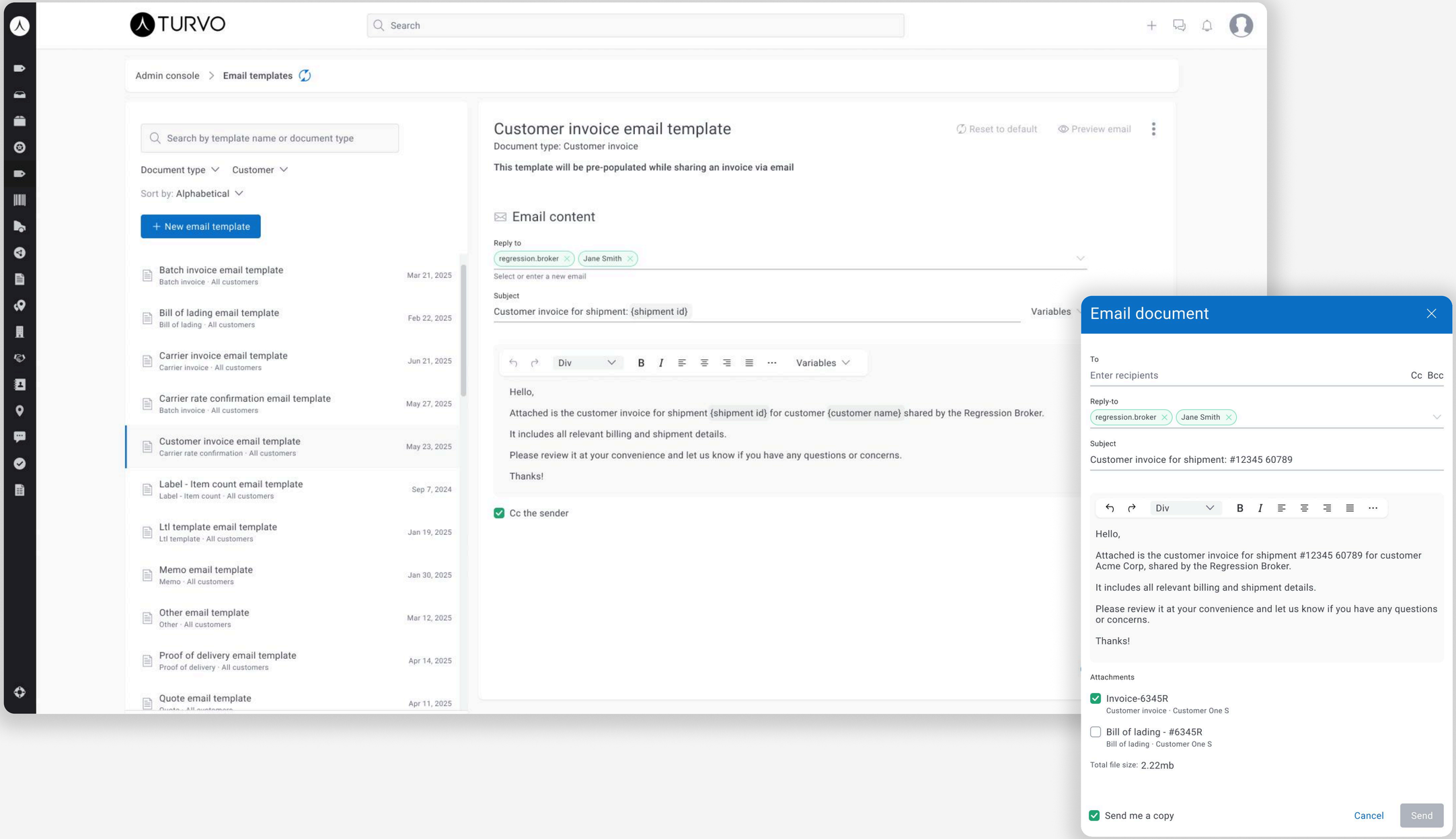
Implementation Support: Annotated designs and collaborated with developers and QA to ensure accuracy and preserve design intent.

CASE STUDY 3 - EMAIL TEMPLATES

We took a holistic approach to the email template design process by analyzing existing Turvo features of emailing shipment documents, and identifying how email templates could enhance those workflows for users.

As part of this, we also updated existing email flow to ensure users experienced a seamless, end-to-end flow rather than isolated improvements.




TURVO
Search
Admin console
Email templates
Search by template name or document type
Document type
Customer
Sort by: Alphabetical
+ New email template
Batch invoice email template
Batch invoice · All customers
Mar 21, 2025
Bill of lading email template
Bill of lading · All customers
Feb 22, 2025
Carrier invoice email template
Carrier invoice · All customers
Jun 21, 2025
Carrier rate confirmation email template
Batch invoice · All customers
May 27, 2025
Customer invoice email template
Carrier rate confirmation · All customers
May 23, 2025
Label - Item count email template
Label - Item count · All customers
Sep 7, 2024
Ltl template email template
Ltl template · All customers
Jan 19, 2025
Memo email template
Memo · All customers
Jan 30, 2025
Other email template
Other · All customers
Mar 12, 2025
Proof of delivery email template
Proof of delivery · All customers
Apr 14, 2025
Quote email template
Apr 11, 2025
Customer invoice email template
Document type: Customer invoice
This template will be pre-populated while sharing an invoice via email
Reset to default
Preview email
Email content
Reply to
regression.broker
Jane Smith
Select or enter a new email
Subject
Customer invoice for shipment: {shipment id}
Variables
Div
Variables
Hello,
Attached is the customer invoice for shipment {shipment id} for customer {customer name} shared by the Regression Broker.
It includes all relevant billing and shipment details.
Please review it at your convenience and let us know if you have any questions or concerns.
Thanks!
Cc the sender
Email document
To
Enter recipients
Cc Bcc
Reply-to
regression.broker
Jane Smith
Subject
Customer invoice for shipment: #12345 60789
Div
Hello,
Attached is the customer invoice for shipment #12345 60789 for customer Acme Corp, shared by the Regression Broker.
It includes all relevant billing and shipment details.
Please review it at your convenience and let us know if you have any questions or concerns.
Thanks!
Attachments
Invoice-6345R
Customer invoice · Customer One S
Bill of lading - #6345R
Bill of lading · Customer One S
Total file size: 2.22mb
Send me a copy
Cancel
Send

ACCOMPLISHMENTS

✅ **Empowered customers** to manage their email templates end-to-end without relying on support.

✅ **Enhanced communication accuracy** with better routing, personalization, and preview tools.

✅ **Supported multiple templates** per document type, enabling flexibility for different customers and modes.

✅ **Improved template discoverability** with robust search, sort, and filter options.

✅ **Strengthened brand presence and professionalism** in all customer-facing communications.

✅ **Cleaned up backend templates** to improve visibility, ensuring previews matched actual emails and reducing confusion and support tickets.

REFLECTION & LEARNINGS

This project highlighted the importance of designing for flexibility and scale, especially in B2B platforms where customer needs vary widely.

Empowering customers to manage their own communication systems not only reduces internal effort but also builds trust and strengthens relationships.

We rolled out the feature in multiple phases due to its size and complexity, allowing customers to benefit from early improvements while giving us time to gather feedback, iterate, and scale the solution gradually.

BEYOND THE CASE STUDIES

Beyond these three projects, I've contributed to several complex initiatives at Turvo, including appointment scheduling, shipments, LTL, dashboards, roles & permissions, rules, and many more. These experiences deepened my understanding of enterprise workflows, scalability, and user empowerment, areas I'd be happy to discuss.

I'm also actively exploring Agentic AI and other AI-driven UX tools. I've experimented with Lovable.ai and v0, using prompts to generate screens and features as a way to understand how these tools might shape future design workflows. This side learning helps me stay familiar with emerging technologies, even if they aren't part of my core toolkit yet.

Continuous learning is central to how I grow as a designer. I regularly engage with platforms like Coursera, attend design and innovation workshops, and explore everyday experiences that broaden my perspective.

THANK YOU

mayuri.amrita@gmail.com | linkedin.com/in/amritamayuri/