

## Project Overview

Shortly after joining the **University of San Diego** as the *Career Events & Marketing Coordinator*, I was tasked with supporting several major campus events. This included the *Fall and Spring Career Fairs*, the *Majors & Minors Fair*, and the *Summer Internship Award* program. What I found was a scattered collection of task lists, files, and institutional memory that lived in email threads, disconnected Notion pages, and most often in the heads of the people who had run these events before.

As a new employee trying to understand how to prepare, coordinate, and execute high-impact events, the experience was unnecessarily stressful and inefficient. There was no documentation, no process ownership, and no scalable way to improve event execution over time. I saw this as a user experience failure, not just for myself, but for my colleagues, stakeholders, and any future hires.

So I took the initiative to build **a department-wide project management system in Notion**. Designed around structure, clarity, and collaboration, the system broke down each event by timeline, task ownership, and context. Which created a blueprint anyone could follow. To complement this, I also designed **a “One Source of Truth” dashboard** for each event. This is meant to be a centralized hub that houses everything from logistics and marketing to historical data and internal roles.

These systems didn't just make project work easier, they made it transparent, repeatable, and continuously improvable. While I was originally asked to contribute to project execution, I went far beyond my role. I created a sustainable, UX-driven infrastructure that now supports every major event our department runs.

## The Problem

When I joined the department, I inherited years' worth of task lists in Notion that, at first glance, seemed like a “project management system.” But it became clear very quickly that they weren't systems at all. they were more like personalized to-do lists built by and only for the person who created them. This meant that if someone left the role or went on vacation, there was no way for another team member to seamlessly take over.

There was no structure, no shared documentation, and no context behind each task. In many cases, we were digging through old email threads to answer basic questions like, “*How many chairs did we order last year?*” or “*What did that outreach email sound like?*” If we needed to make a room reservation or contact a vendor, there were no instructions. Everything was passed on verbally making the knowledge inefficiently scattered. So much depended on knowing the right person to ask, and hoping they replied with enough detail. This led to constant bottlenecks, wasted time, and frustration. I spent more time trying to decipher what a task actually meant than completing it. Navigating from Notion to Google Drive to online university portals to someone’s half-remembered explanation became the norm.

As a new employee, this disorganization made onboarding incredibly inefficient. Instead of learning how our systems worked, I had to reverse-engineer workflows and decode fragmented information. I quickly saw that I wasn’t alone. A majority of team members were stuck in a loop of recreating work that had already been done, simply because it wasn’t documented or accessible. Ironically, years of archived data made things worse. Searching “Fall Fair” in Notion brought up an overwhelming mess of unstructured notes, outdated task lists, and duplicate content. Something along the digital equivalent of a cluttered filing cabinet with no labels. I realized this wasn’t just my problem. **It was a systemic failure that made everyone’s work harder.** I couldn’t help but feel this was holding the department back from improving, standardizing, and scaling. So I decided to fix it not just for myself, but for the team now and in the future.

## **The Solution Part A - Project Management System**

I set out to build a system that was everything the previous one wasn’t: user-friendly, structured, organized, and instantly understandable. My core principle was eliminating inefficiency. I wanted anyone, a new employee, a team member, or a stakeholder, to be able to look at the system and immediately know a few things. What needs to be done, when it needs to happen, and how to go about doing it.

A major frustration I encountered early on was the culture of *institutional knowledge*. Where critical information lived in people’s heads and required constant follow-ups. I saw this as a form of unintentional gatekeeping. I

wanted to replace that with a system rooted in openness, clarity, and self-sufficiency.

The system I designed in Notion introduced:

- A top-to-bottom timeline structure: Tasks are organized chronologically, making it intuitive to know what's upcoming without needing additional explanation.
- Consistent task templates: Every task follows the same internal structure: Who, What, When, Where, Why. I hoped this would remove cognitive friction. Once users open two tasks, they know how to use the rest.
- Clear ownership and roles: Each task identifies a lead, supporting contacts, and relevant links or platforms. Yay, no more digging or guessing.
- Built-in documentation: Instructional content is embedded directly inside tasks, reducing the need for training or outside context.
- Flexible usability: Users can filter by due date, category, or status or simply scroll through and visually scan based on position. The system adapts to whatever method is most intuitive to them.

This wasn't about building a tool that worked for me. It was about designing a system that worked for everyone. Whether someone was planning the entire event or helping with one detail, they could quickly find what they needed and move forward without bottlenecks, meetings, or confusion.

Compared to the old system, which required interpretation and frequent back-and-forth just to complete basic tasks, this version feels like a night-and-day shift. Now, team members could jump in and collaborate with confidence. No special training, no second guessing, no chaos.

## The Solution Part B - One Source of Truth Dashboard

While the new project management system solved many workflow issues within our four-person team, it quickly became clear that the rest of the 18 person department required a bit more. Most didn't even know how to use Notion, let alone how our internal system worked. That meant project knowledge wasn't just limited, it was siloed to our design team. If someone outside our team needed background on a recurring event, they had one option: email the person who led it last year and hope they responded with enough information.

That is why I also built the **One Source of Truth Dashboard** to eliminate that dependency. Its purpose was simple: give everyone access to the information they need without needing to ask for it.

The dashboard is organized into six core areas, each housed in a visual board layout:

- Event Overview – Purpose, who it serves, and why it matters
- Marketing Overview – Flyer timelines, campaign links, branding guides
- Logistics Overview – Vendor contacts, inventory systems, on-the-ground coordination
- Key Resources – Reports, templates, forms, and past documentation
- Outcomes Overview – Attendance data, feedback, impact stories
- Team Notes – Insider knowledge, common mistakes, pro tips for future teams.

Each section follows the same consistent structure: Purpose, Overview, Details, Contacts, Tips, and a Notice. That way, users don't have to re-learn the system every time they open a new section. The content is detailed, clearly labeled, categorized by year, and deeply UX-oriented.

The dashboard supports a wide range of users:

- New employees learning the ropes
- Longtime team members looking to streamline their work
- Stakeholders needing accurate, centralized info
- Anyone trying to understand what was done and why it mattered

It also allows us to capture institutional wisdom that's usually lost over time. Cool tips like *"don't forget to ask facilities to shut off sprinklers"* or reflections on what worked and didn't. Instead of being buried in inboxes or memory, that knowledge lives in an accessible, structured system for future teams to build on.

Now, I must be honest about something. Was this officially adopted by leadership? No, but I adopted it for myself and the future person in my role. And that matters. If someone were to step into this job tomorrow, they could spend one afternoon reading through the dashboard and walk away with more relevant knowledge than someone who's been here 20 years. That's the power of a well-designed knowledge system.

## **Outcome**

The new system had an immediate impact on both my work and the broader team dynamic. On a personal level, building this structure forced me to research every process tied to our events, which helped me onboard faster than any formal training could have. If it had been implemented, the clarity and consistency it would have provided in eliminating the guesswork from daily responsibilities would without a doubt have improved the entire department. People could trust that they wouldn't miss a single task or deadline, and could work proactively rather than reactively.

Team members noticed. Everyone I showed it to echoed the same sentiment: it was clear, easy to follow, and incredibly detailed. While only a few had the chance to explore the "one source of truth" dashboard deeply, even a quick glance sparked genuine surprise at the depth of information included. My manager audibly gasped when he saw it. Which I believe was a testament to how much clearer things could be. Ironically, long-time employees began

asking me for information, despite me being one of the newest people on the team.

The potential for onboarding is huge. With these tools in place, a brand-new employee could be brought up to speed faster than ever. Not just to understand what's been done, but how and why. It's not an exaggeration to say that reading through these systems could make someone an expert in a matter of hours.

But the biggest outcome of all was personal. This project didn't just prove my ability to think and work like a UX designer, it made me realize I am one. I didn't need motivation to build these systems because every moment of doing it was energizing. I loved taking something broken (or nonexistent) and crafting something useful, structured, and beautiful. **That's what UX is at its core: making things work better for users.**

## Reflection + Lessons Learned

This project was more than just a system build. It was a moment of clarity. I realized how deeply I value structure, transparency, and usability in every interaction. Despite never having used Notion before this role, I designed and implemented a fully structured project management system and a scalable reference dashboard within weeks of starting because the existing "system" (or lack thereof) caused real, unnecessary stress for me and others.

I learned that UX isn't confined to interfaces or apps, it's everywhere. It lives in onboarding, operations, team communication, and internal knowledge management. If a person needs to jump between platforms, search through years of disorganized emails, or ask five different people to get one answer, that's a user experience failure. Fixing that became my obsession.

If I had more time, I would've conducted deeper research with my team to map exactly how many steps or blockers they experienced before these systems were implemented. I was focused on solving the problems I personally encountered as a new employee but now I recognize how valuable it would've been to quantify and visualize those inefficiencies across the team. That would've helped me tell an even clearer UX story with data.

This work also reshaped how I view onboarding. I now understand just how impactful good UX can be in internal operations. Not just for end users, but for the people behind the scenes.

The dashboard and task system I built could become core onboarding materials. In fact, I strongly already advocated for this to happen and I even created a fun UX quiz for the department to raise awareness.

In the end, I was proud not only of the work, but of how I felt while doing it. I hit a deep state of focus, joy, and energy. The same way I feel when I'm coding. It never felt like "*work*." I was building, problem-solving, and designing for people and that's when I knew for sure: **I am a UX designer.**