کیس استادی

آهن پیمان

طراحی یک سیستم یکپارچه برای مدیریت تولید کارخانه

+۵۰ صفحه
+۸۰ کامپونت
استایل گاید
داشبورد مدیریت

امید مددی طراح محصول

درباره آهن پیمان

آهن پیمان یک سامانه تحت وب برای مدیریت فرایندهای کارخانه است؛ از تعریف پروژه و برنامه‌ریزی تولید تا کنترل کیفیت، انبار، بسته‌بندی و ارسال.
این پروژه را طی دو ماه و به‌عنوان تنها طراح محصول انجام دادم. مسئولیت من تمام مراحل طراحی محصول، از شناخت مسئله و ترسیم جریان‌ها تا وایرفریم، طراحی رابط کاربری و ایجاد کامپوننت‌ها بود.

پلتفرم: **تحت وب**
نقش من: **طراح محصول**
حوزه محصول: **Manufacturing Management System**
مدت پروژه: **دو ماه**
وضعیت پروژه: **طراحی کامل شد، اما وارد مرحله توسعه نشد**

Problem

مسئله‌ای فراتر از ثبت قطعه

در کارخانه، هر قطعه فقط یک عدد در موجودی نیست. باید مشخص باشد به کدام پروژه و زیرپروژه تعلق دارد، چه عملیات‌هایی را پشت سر گذاشته، چه کسی و با چه دستگاهی آن را تولید کرده، آیا از کنترل کیفیت عبور کرده و اکنون در کدام محل یا بسته قرار دارد. در مرحله‌ی ارسال نیز مسیر، مقصد، وسیله‌ی نقلیه و اسناد مرتبط باید قابل ردیابی باشند.

پیش از طراحی، این اطلاعات و تصمیم‌ها میان واحدهای مختلفی مانند مدیریت پروژه، تولید، کنترل کیفیت، انبار و حمل‌ونقل توزیع شده بود. هر واحد بخشی از واقعیت را می‌دید، اما تصمیم‌های آن‌ها به هم وابسته بود. بنابراین چالش اصلی، ساختن یک رابط زیبا نبود؛ باید منطق مشترکی می‌ساختم که ارتباط مراحل را برای همه‌ی نقش‌ها روشن کند.

Goal

هدف طراحی

طراحی یک سامانه تحت وب که ثبت فرایندهای تولید، مدیریت پیشرفته‌ی انبار و گزارش‌گیری را در یک تجربه‌ی منسجم فراهم کند؛ بدون آن‌که کاربر در جزئیات غیرضروری گم شود.

ROLE & APPROACH

نقش و مسئولیت من

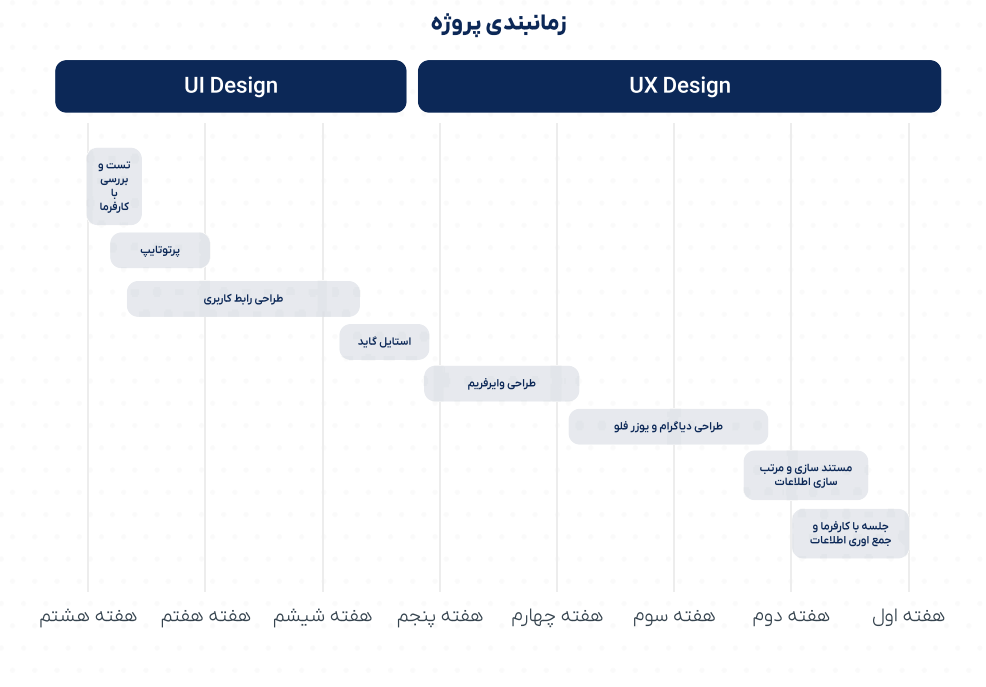
به‌عنوان تنها طراح محصول پروژه، مسئولیت کل مسیر طراحی از فهم مسئله تا آماده‌سازی خروجی برای توسعه بر عهده‌ی من بود.

- ترسیم فرایندها و User Flowها
- برگزاری و مستندسازی جلسات شناخت مسئله
- تعریف الگوهای تعامل و وضعیت‌ها
- طراحی معماری اطلاعات
- طراحی کامپوننت‌ها و الگوهای تکرارشونده
- دسته‌بندی نیازهای کسب‌وکار و نقش‌ها
- مرورهای مستمر با مدیرعامل و رئیس IT
- تحویل طراحی نهایی ماژول‌ها
- وایرفریم و طراحی رابط کاربری
- آماده‌سازی سناریوهای خطا و تأیید

شناخت اولیه از طریق جلسات با مدیرعامل و رئیس واحد IT انجام شد. گفت‌وگوهای پراکنده‌ی عملیاتی را ثبت و به ساختاری قابل‌بحث تبدیل کردم تا پیش از ورود به جزئیات بصری، درباره‌ی روابط میان واحدها توافق داشته باشیم.

Scheduling

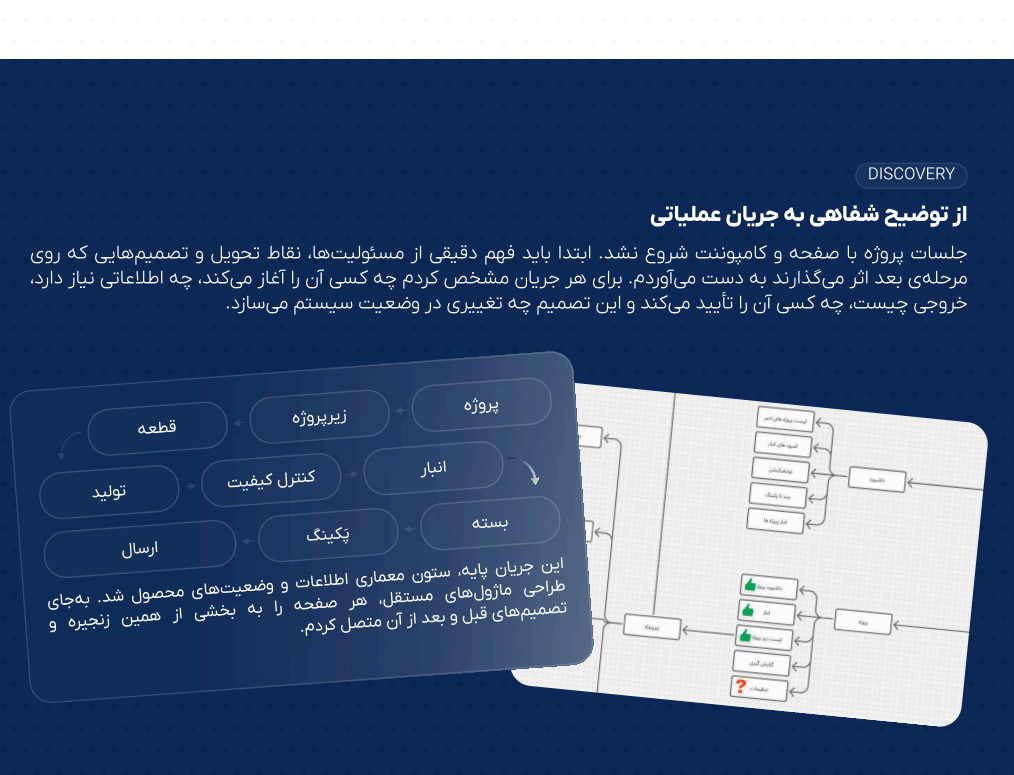
زمانبندی پروژه



DISCOVERY

از توضیح شفاهی به جریان عملیاتی

جلسات پروژه با صفحه و کامپوننت شروع نشد. ابتدا باید فهم دقیقی از مسئولیت‌ها، نقاط تحویل و تصمیم‌هایی که روی مرحله‌ی بعد اثر می‌گذارند به دست می‌آوردم. برای هر جریان مشخص کردم چه کسی آن را آغاز می‌کند، چه اطلاعاتی نیاز دارد، خروجی چیست، چه کسی آن را تأیید می‌کند و این تصمیم چه تغییری در وضعیت سیستم می‌سازد.



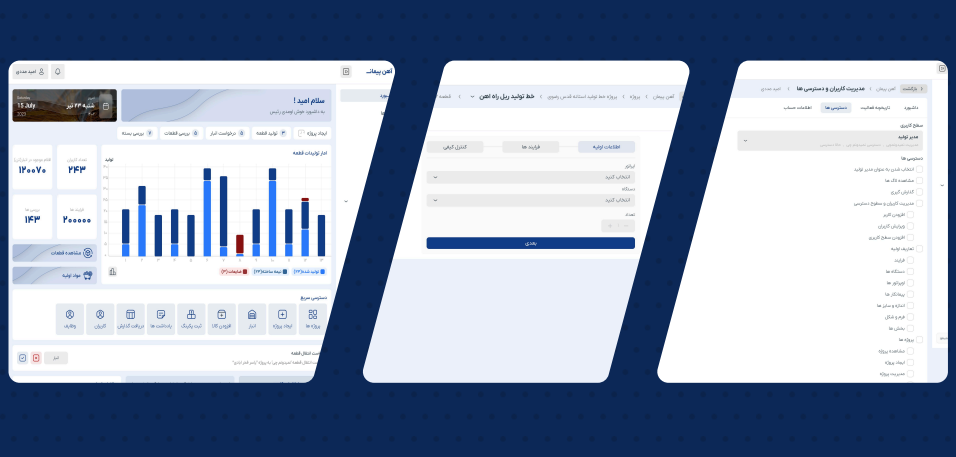
این جریان پایه، ستون معماری اطلاعات و وضعیت‌های محصول شد. به‌جای طراحی ماژول‌های مستقل، هر صفحه را به بخشی از همین زنجیره و تصمیم‌های قبل و بعد از آن متصل کردم.

Roles

نقش‌ها، یک محصول، نیازهای متفاوت

مدیر پروژه به وضعیت کلی، پیشرفت و ریسک‌ها نیاز داشت؛ اپراتور باید سریع و دقیق ثبت می‌کرد؛ مسئول QC باید موارد در انتظار بررسی را می‌دید؛ و انباردار با موجودی، درخواست‌ها و جابه‌جایی‌ها کار می‌کرد. یک داشبورد و سطح دسترسی یکسان، برای این نیازهای متفاوت مناسب نبود.

تصمیم طراحی: ساختار کلی سیستم ثابت ماند، اما محتوای داشبورد، دسترسی‌ها و اقدام اصلی بر اساس نقش تغییر کرد. این انتخاب مسیر یادگیری محصول را حفظ کرد و در عین حال، اطلاعات هر نقش را مرتبط نگه داشت.

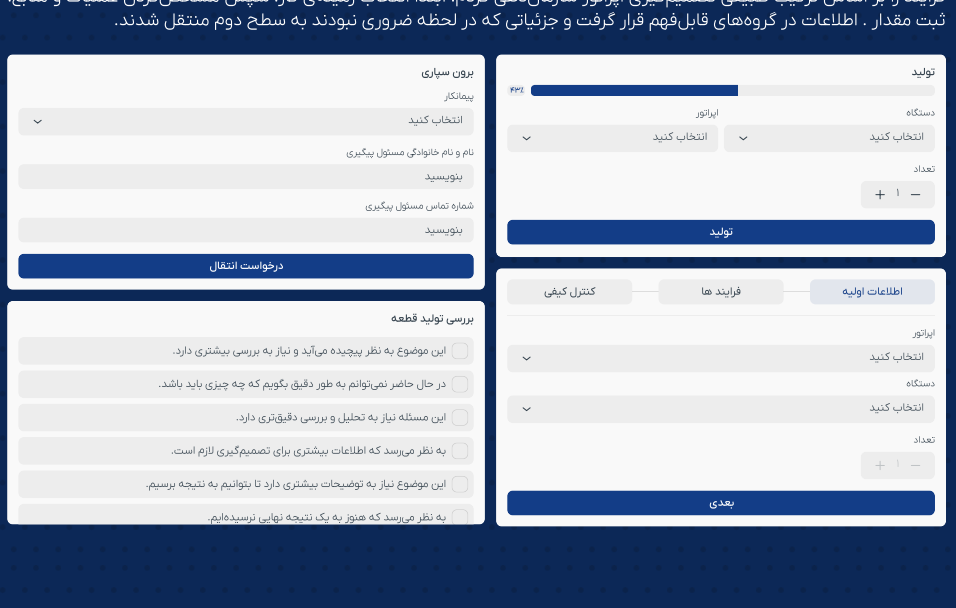


DESIGN DECISIONS 1

تصمیم ۱: ثبت تولید مطابق ترتیب ذهنی کاربر

ثبت تولید به داده‌های زیادی وابسته بود: پروژه، زیرپروژه، قطعه، عملیات، اپراتور، دستگاه، مقدار . نمایش همزمان تمام فیلدها، فرم را سنگین و مستعد خطا می‌کرد.

فرایند را بر اساس ترتیب طبیعی تصمیم‌گیری اپراتور سازمان‌دهی کردم: ابتدا انتخاب زمینه‌ی کار، سپس مشخص‌کردن عملیات و منابع، و در نهایت ثبت مقدار . اطلاعات در گروه‌های قابل‌فهم قرار گرفت و جزئیاتی که در لحظه ضروری نبودند به سطح دوم منتقل شدند.

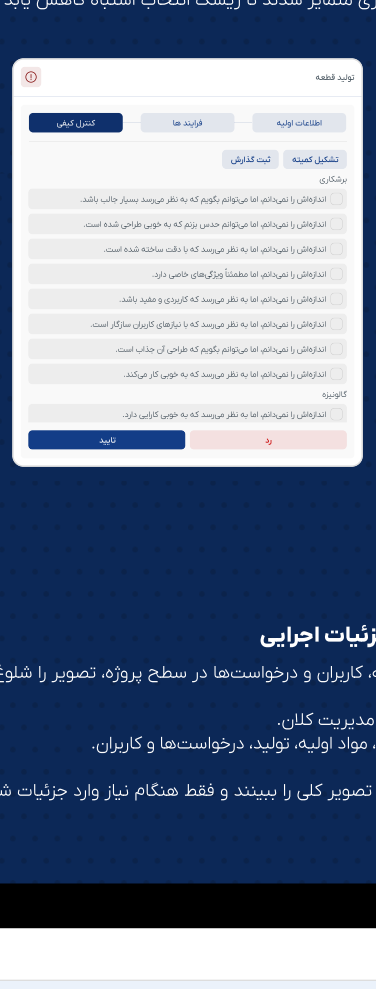


DESIGN DECISIONS 2

تصمیم ۲: کنترل کیفیت به‌عنوان نقطه‌ی تصمیم

QC صرفاً ثبت یک نتیجه نبود. تأیید یا رد قطعه مشخص می‌کرد که آیا قطعه می‌تواند وارد انبار و بسته‌بندی شود یا باید به مرحله‌ای قبل بازگردد. بنابراین همه‌ی اطلاعات لازم برای تصمیم‌گیری را در یک صفحه جمع کردم: مشخصات قطعه، پروژه، سابقه‌ی تولید، اپراتور و دستگاه، چک‌لیست، نتیجه، علت رد و اقدام بعدی.

اقدام‌های تأیید و رد از نظر بصری و رفتاری متمایز شدند تا ریسک انتخاب اشتباه کاهش یابد و نتیجه‌ی هر تصمیم در جریان محصول قابل مشاهده بماند.



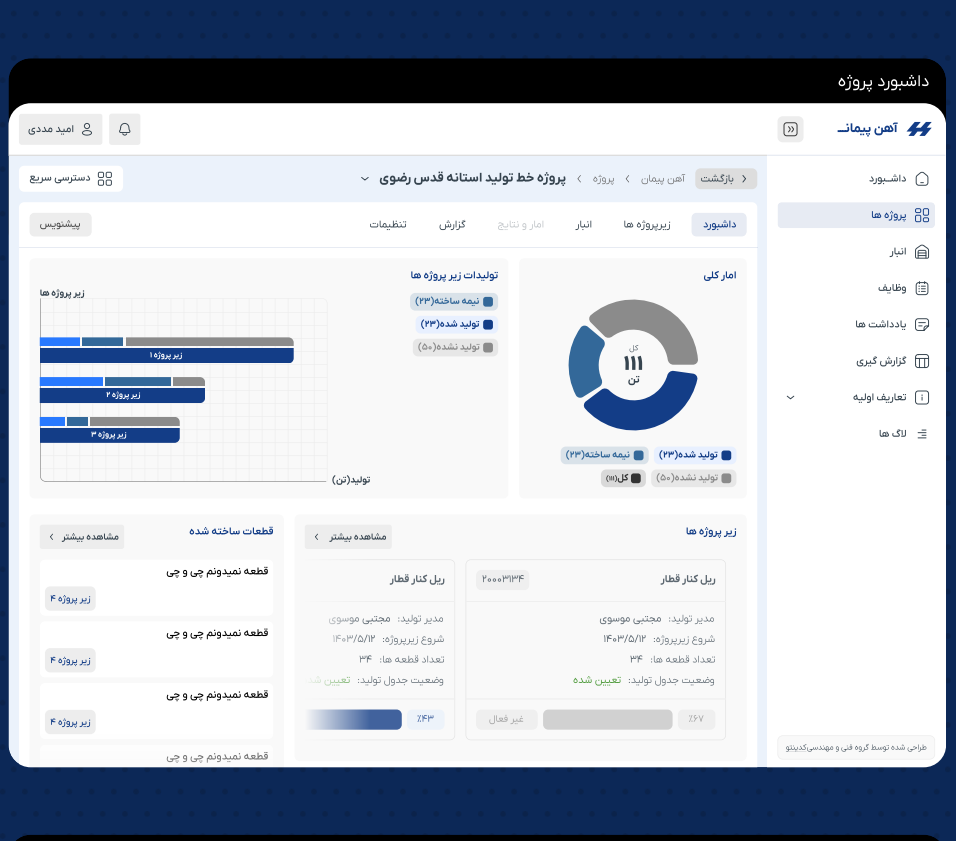
DESIGN DECISIONS 3

تصمیم ۳: تفکیک نمای کلی و جزئیات اجرایی

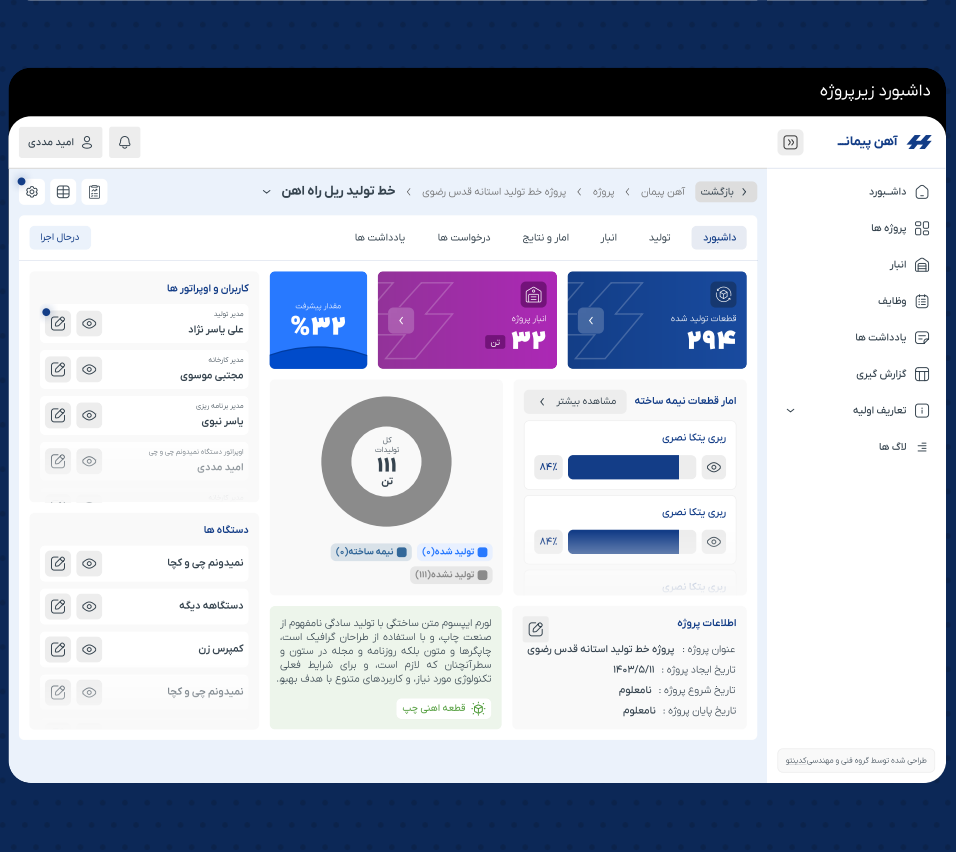
قرار دادن تمام اطلاعات تولید، مواد اولیه، کاربران و درخواست‌ها در سطح پروژه، تصویر را شلوغ می‌کرد. پروژه و زیرپروژه را به دو سطح معنادار تقسیم کردم:

- پروژه: وضعیت کلی، پیشرفت، آمار و مدیریت کلان.
- زیرپروژه: جزئیات اجرایی مانند BOM، مواد اولیه، تولید، درخواست‌ها و کاربران.

این تفکیک به مدیران اجازه می‌داد ابتدا تصویر کلی را ببینند و فقط هنگام نیاز وارد جزئیات شوند؛ در نتیجه، ناوبری و اولویت اطلاعات در هر سطح روشن‌تر شد.



داشبورد پروژه

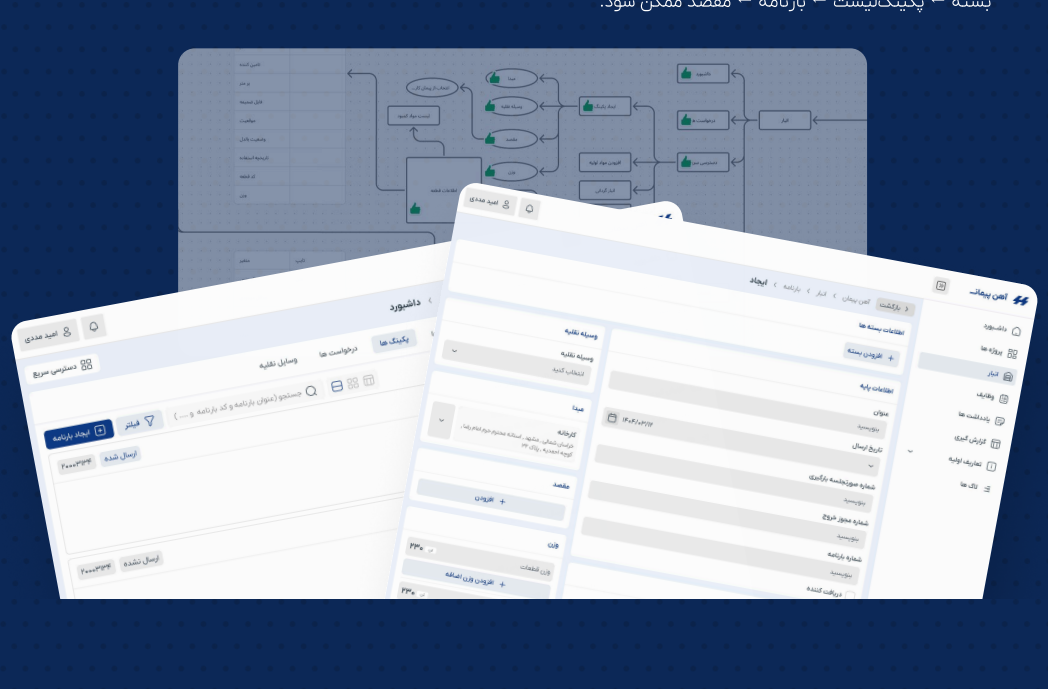


داشبورد زیرپروژه

DESIGN DECISIONS 4

تصمیم ۴: بسته به‌عنوان یک موجودیت مستقل

در بخش انبار و ارسال، بسته صرفاً مجموعه‌ای از قطعات نبود. هر بسته پروژه و زیرپروژه‌ی مرتبط، محتویات، وزن فعلی و سقف وزن، وضعیت، پکینگ‌لیست و تاریخچه‌ی تغییرات خود را داشت. آن را به یک موجودیت مستقل تبدیل کردم تا ردیابی زنجیره‌ی قطعه ← بسته ← پکینگ‌لیست ← بارنامه ← مقصد ممکن شود.

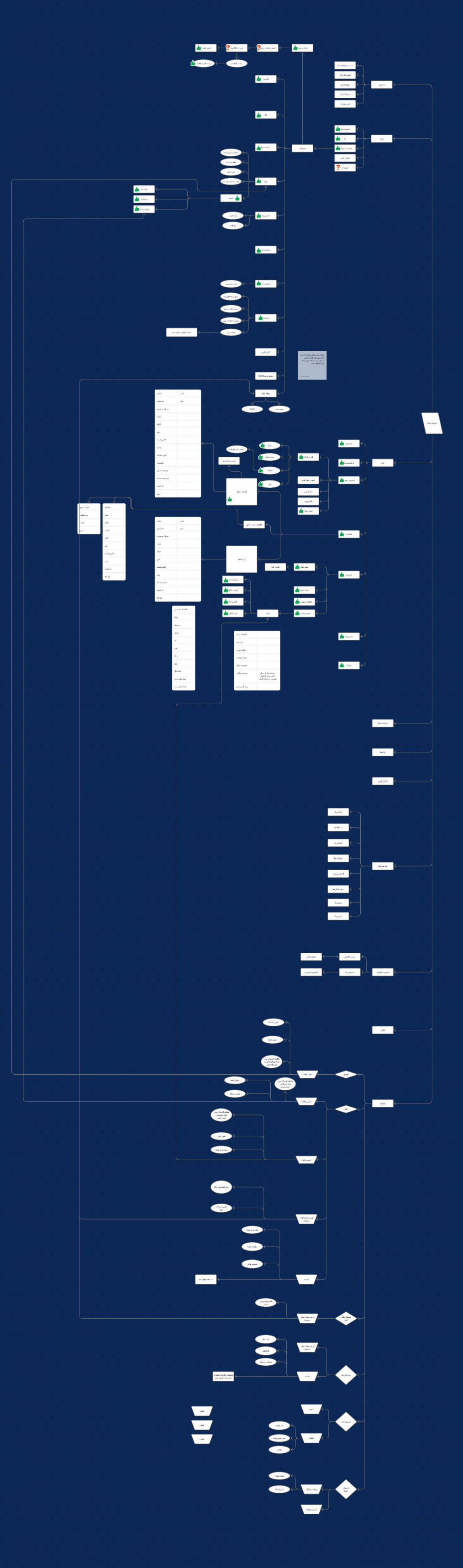


INFORMATION ARCHITECTURE

معماری اطلاعات؛ ساختار قبل از صفحه

پس از ترسیم جریان اصلی، ماژول‌ها را به‌صورت یک فهرست جدا از هم طراحی نکردم. هر بخش باید به یک سؤال عملیاتی پاسخ می‌داد و ورودی و خروجی‌اش با مرحله‌ی قبل و بعد مشخص می‌بود. به این ترتیب، «پروژه» زمینه‌ی کار را تعریف می‌کرد، «زیرپروژه» جزئیات اجرایی را نگه می‌داشت، و هر قطعه در طول تولید، QC، انبار و ارسال یک تاریخچه‌ی پیوسته داشت.

این مدل کمک کرد از دو مشکل رایج جلوگیری کنم: تکرار اطلاعات در چند بخش و نمایش مسیرهایی که کاربر نمی‌داند از کجا شروع و در کجا تمام می‌شوند.



FLOW 01 — PRODUCTION

فلو ثبت تولید

ثبت تولید یکی از پرتکرارترین و حساس‌ترین عملیات سامانه بود. اپراتور باید در کمترین زمان، قطعه‌ی درست را در بستر پروژه‌ی درست انتخاب می‌کرد، عملیات و منابع را مشخص می‌کرد و خروجی را به مرحله‌ی بعد می‌فرستاد. اشتباه در هرکدام از این انتخاب‌ها می‌توانست ردیابی قطعه را در کل زنجیره مختل کند.

Problem

مسئله

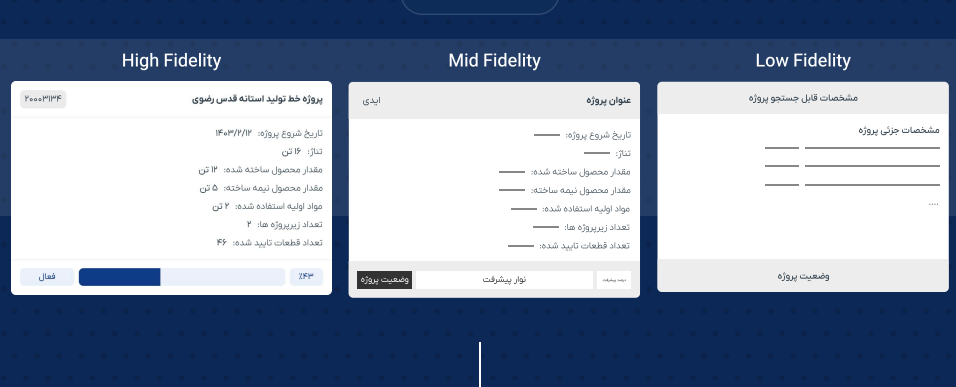
در فرم‌های اولیه، حجم داده و وابستگی فیلدها زیاد بود. اگر اطلاعات پروژه، قطعه، عملیات، اپراتور و دستگاه همزمان و بدون ترتیب نمایش داده می‌شدند، بار شناختی بالا می‌رفت و احتمال خطا بیشتر می‌شد.

Solution

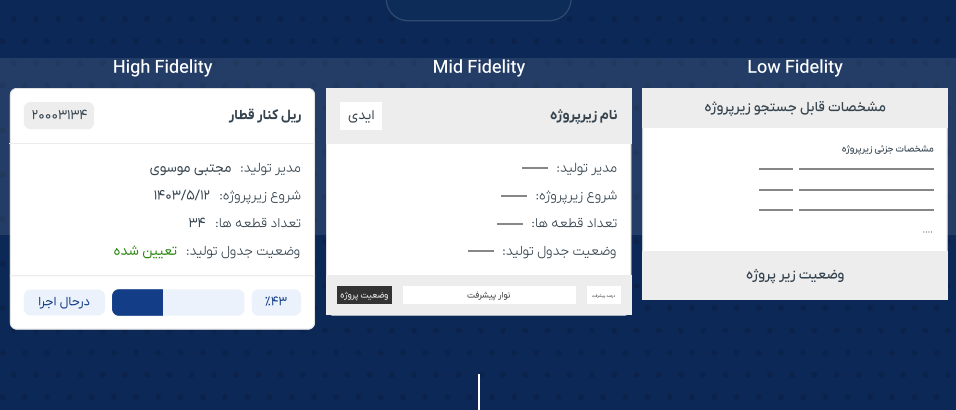
راه‌حل

فرم را به ترتیب تصمیم‌گیری واقعی کاربر تقسیم کردم: انتخاب پروژه و زیرپروژه، انتخاب قطعه، تعیین عملیات، انتخاب اپراتور و دستگاه، ثبت مقدار و در پایان تعیین مقصد یا مرحله‌ی بعد. هر انتخاب، زمینه‌ی گزینه‌های بعدی را محدود می‌کرد تا کاربر فقط با گزینه‌های معتبر روبه‌رو باشد.

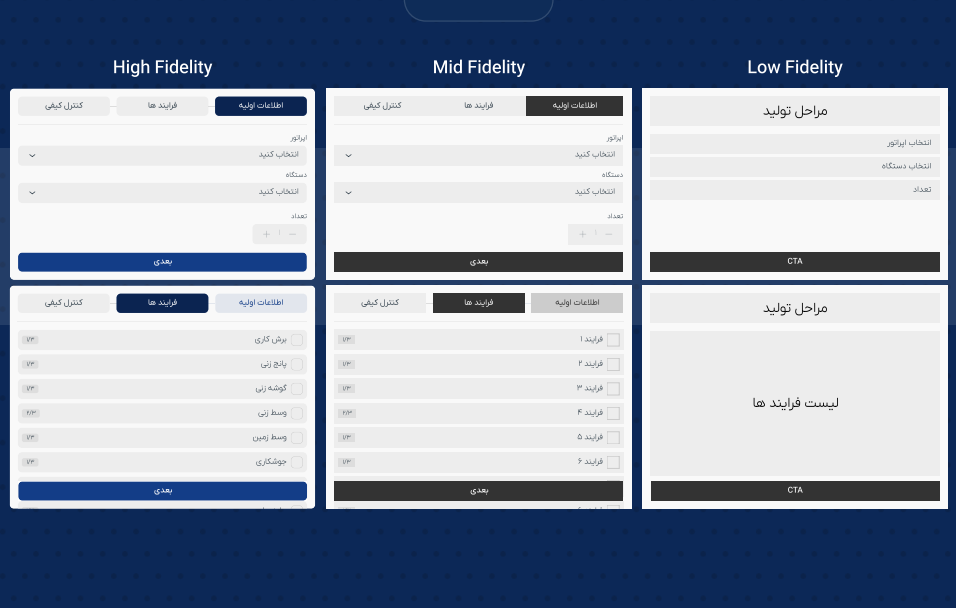
انتخاب پروژه



انتخاب زیرپروژه

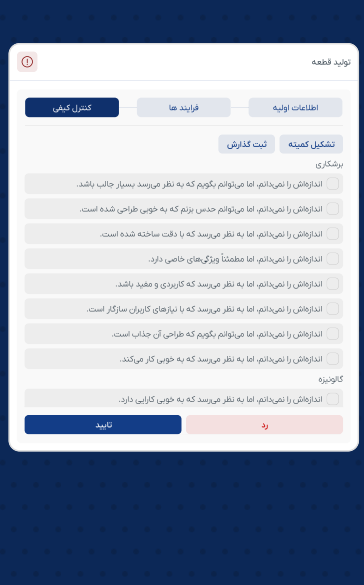


تولید قطعه



FLOW 02 — QUALITY CONTROL

کنترل کیفیت؛ دروازه‌ی تغییر وضعیت



نتیجه‌ی QC فقط یک داده‌ی ثبت‌شده نبود؛ یک تصمیم عملیاتی بود که تعیین می‌کرد قطعه وارد انبار و بسته‌بندی می‌شود یا به مرحله‌ای قبل بازمی‌گردد. این وابستگی، صفحه‌ی کنترل کیفیت را به نقطه‌ای حساس در تجربه‌ی محصول تبدیل می‌کرد.

اطلاعات لازم برای تصمیم

برای جلوگیری از جابه‌جایی میان صفحه‌ها، مشخصات قطعه، پروژه و زیرپروژه‌ی مرتبط، ثبت تولید، اپراتور و دستگاه، چک‌لیست کنترل، نتیجه، علت رد و اقدام بعدی را در یک مسیر واحد کنار هم قرار دادم.

رفتار تأیید و رد

تأیید و رد هم از نظر رنگ و هم از نظر پیامد رفتاری جدا شدند. در حالت رد، علت و مسیر بازگشت به مرحله‌ی قبل بخشی از تصمیم بود؛ بنابراین چالش اصلی فقط یک دکمه‌ی قرمز نبود، بلکه کاربر باید تصمیم را با دلیل ثبت می‌کرد.

FLOW 04 — WAREHOUSE & SHIPPING

از قطعه‌ی تأییدشده تا مقصد

پس از عبور از QC، قطعات وارد فرایند انبار، بسته‌بندی و ارسال می‌شدند. در این مرحله، ردیابی فقط به موجودی محدود نبود: باید امکان مشاهده‌ی محل قطعه، بسته‌ای که در آن قرار گرفته، وزن بسته، پکینگ‌لیست، بارنامه و مقصد نهایی وجود می‌داشت.

بسته به‌عنوان یک موجودیت مستقل

هر بسته هویت خودش را داشت: پروژه و زیرپروژه‌ی مرتبط، محتویات، وزن فعلی و سقف وزن، وضعیت، پکینگ‌لیست و تاریخچه‌ی تغییرات. این تصمیم به‌جای نگاه‌کردن به بسته به‌عنوان یک ظرف ساده، آن را به حلقه‌ی قابل‌پیگیری میان قطعه و ارسال تبدیل کرد.

اصل تعامل

کاربر ابتدا قطعات تأییدشده را به بسته اضافه می‌کرد، سپس با تکمیل و تأیید بسته وارد پکینگ می‌شد و در مرحله‌ی آخر اطلاعات بارنامه و حمل‌ونقل را ثبت می‌کرد. در هر مرحله فقط اقدام‌های معتبر همان وضعیت نمایش داده می‌شدند.

انبار · بسته · پکینگ‌لیست · بارنامه · مقصد

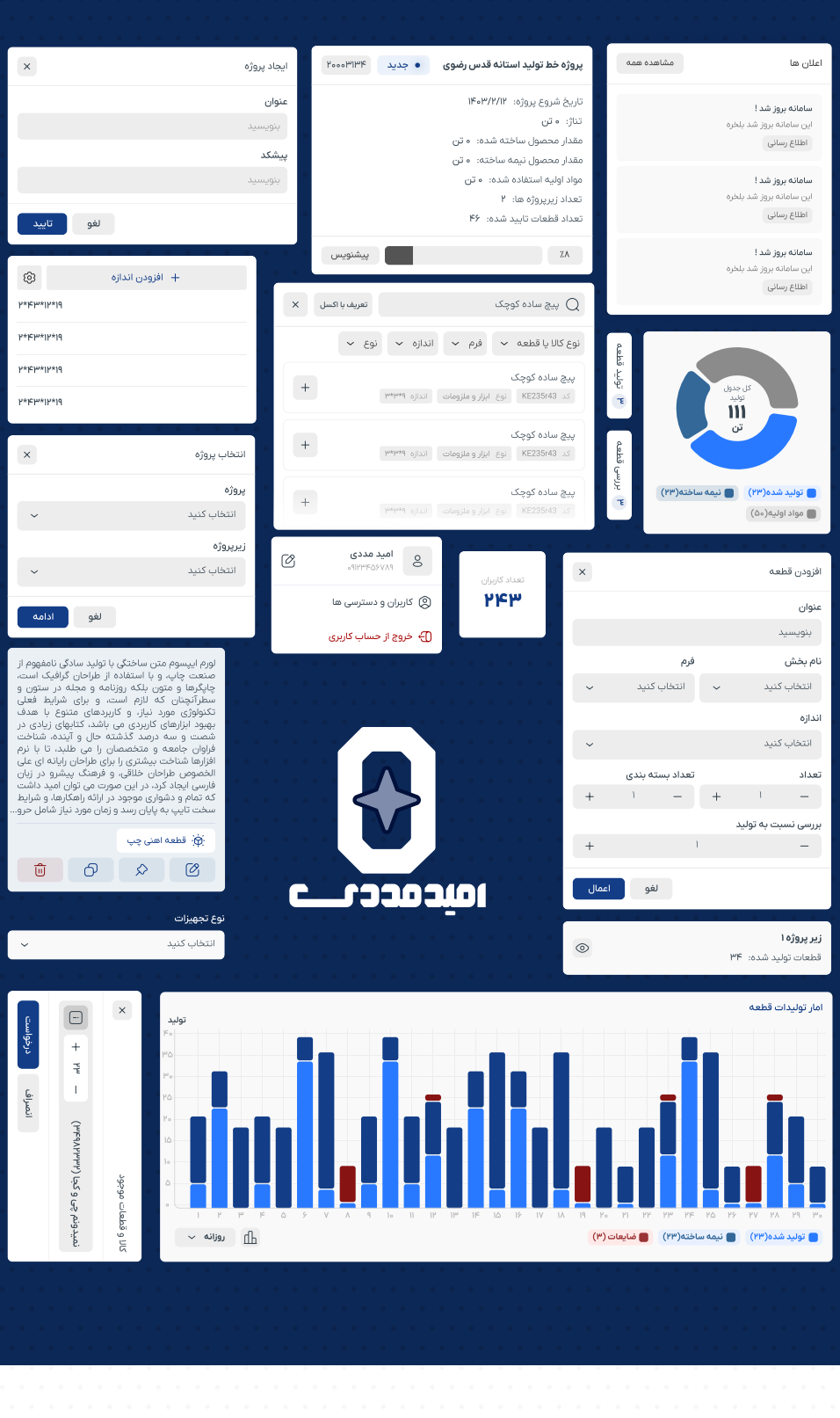
SYSTEM THINKING

طراحی برای حجم بالای اطلاعات

محصول شامل ۱۲ بخش اصلی و تعداد زیادی جدول، فرم، فیلتر و وضعیت بود. برای حفظ یکپارچگی، الگوهای مشترک تعریف کردم تا کاربر پس از یادگیری یک الگو، بتواند رفتار مشابه را در دیگر بخش‌ها پیش‌بینی کند.

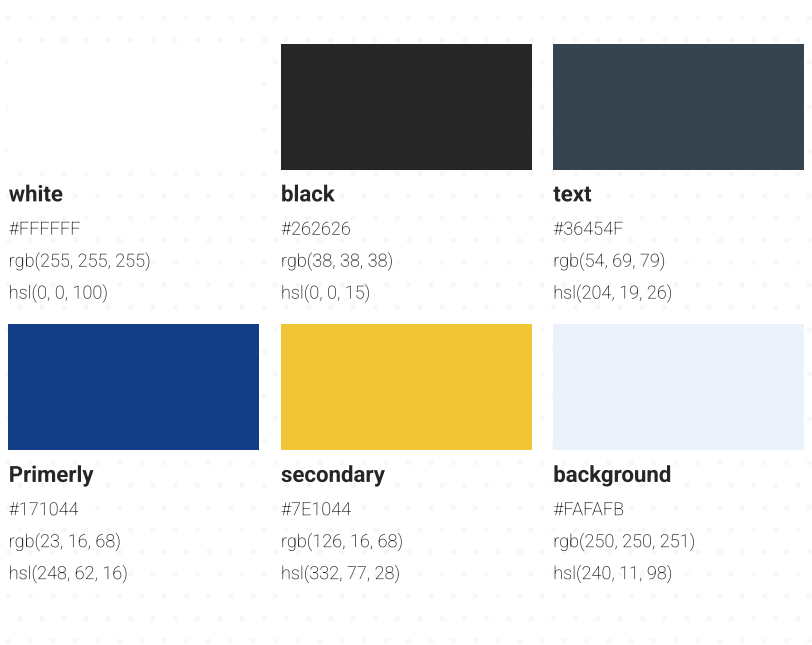
- دسترسی و نقش‌ها
- تاریخچه‌ی فعالیت‌ها
- بارگذاری فایل
- اعلان و یادداشت
- Empty Stateها
- فرم‌ها و اعتبارسنجی
- Modalهای تأیید
- Badgeهای وضعیت و هشدار
- جست‌وجو و فیلتر
- جدول، صفحه‌بندی و عملیات هر ردیف

سیستم طراحی فقط برای هماهنگی ظاهری ساخته نشد؛ هدف، ایجاد رفتار ثابت برای تصمیم‌ها و عملیات‌های تکرارشونده در پروژه، تولید، QC، انبار و گزارش‌ها بود.

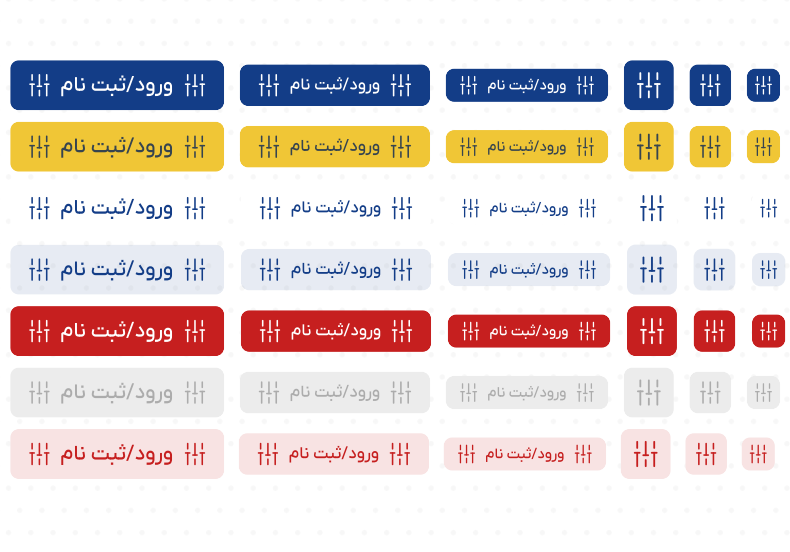


Color palette

پالت رنگی



white	#FFFFFF	rgb(255, 255, 255)	hsl(0, 0, 100)
black	#262626	rgb(38, 38, 38)	hsl(0, 0, 15)
text	#36454F	rgb(54, 69, 79)	hsl(204, 19, 26)
Primerly	#171044	rgb(23, 16, 68)	hsl(248, 62, 16)
secondary	#7E1044	rgb(126, 16, 68)	hsl(332, 77, 28)
background	#FAFAFB	rgb(250, 250, 251)	hsl(240, 11, 98)



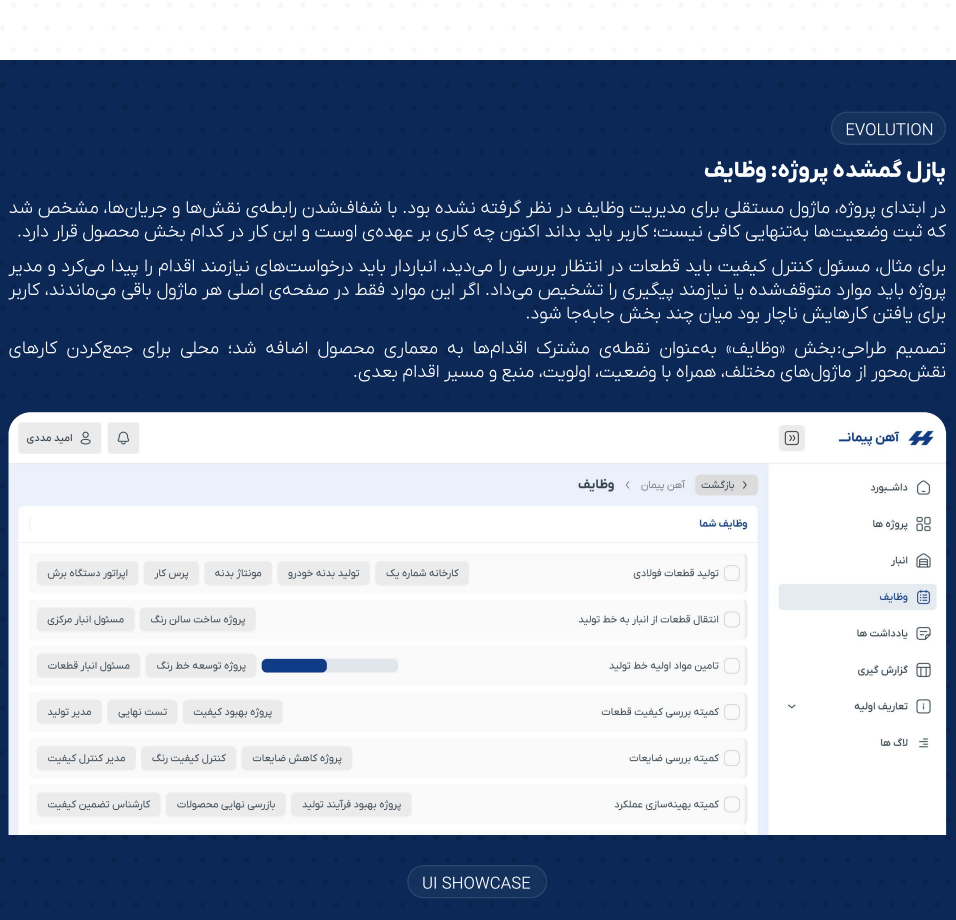
EVOLUTION

پازل گمشده پروژه: وظایف

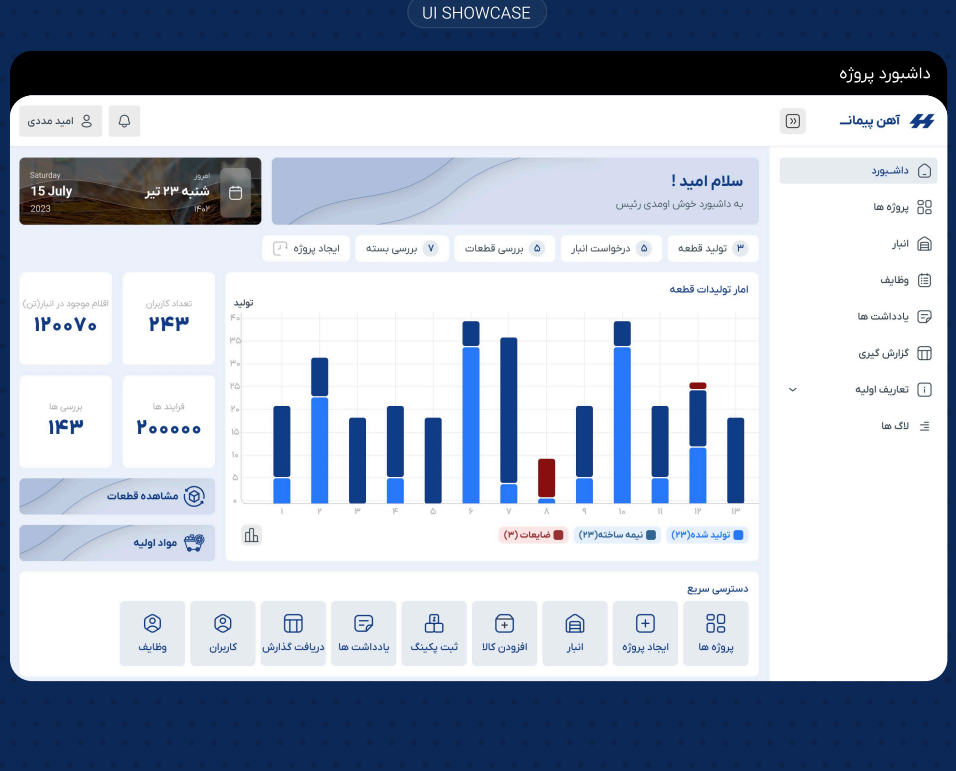
در ابتدای پروژه، ماژول مستقلی برای مدیریت وظایف در نظر گرفته نشده بود. با شفاف‌شدن رابطه‌ی نقش‌ها و جریان‌ها، مشخص شد که ثبت وضعیت‌ها به‌تنهایی کافی نیست؛ کاربر باید بداند اکنون چه کاری بر عهده‌ی اوست و این کار در کدام بخش محصول قرار دارد.

برای مثال، مسئول کنترل کیفیت باید قطعات در انتظار بررسی را می‌دید، انباردار باید درخواست‌های نیازمند اقدام را پیدا می‌کرد و مدیر پروژه باید موارد متوقف‌شده یا نیازمند پیگیری را تشخیص می‌داد. اگر این موارد فقط در صفحه‌ی اصلی هر ماژول باقی می‌ماندند، کاربر برای یافتن کارهایش ناچار بود میان چند بخش جابه‌جا شود.

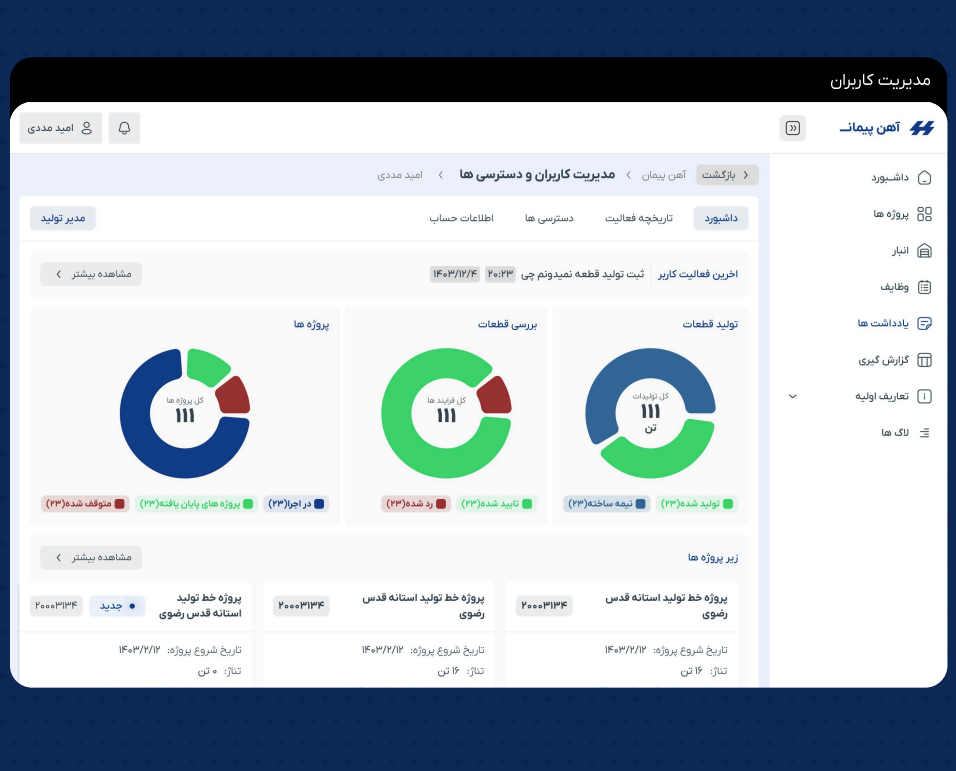
تصمیم طراحی:بخش «وظایف» به‌عنوان نقطه‌ی مشترک اقدام‌ها به معماری محصول اضافه شد: محلی برای جمع‌کردن کارهای نقش‌محور از ماژول‌های مختلف، همراه با وضعیت، اولویت، منبع و مسیر اقدام بعدی.



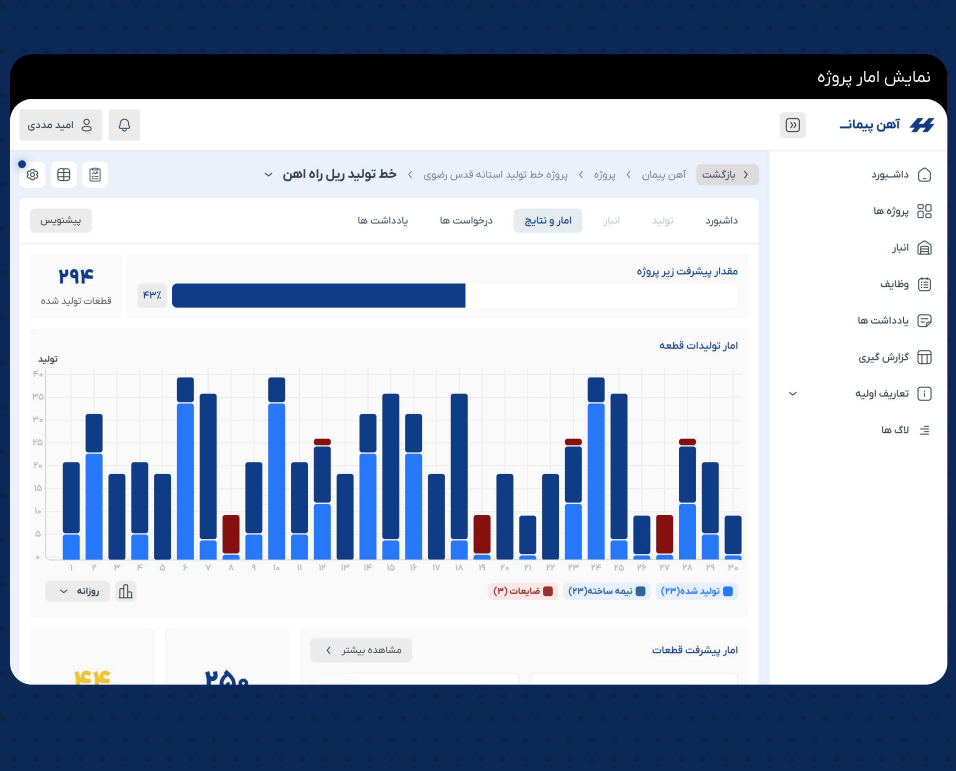
UI SHOWCASE



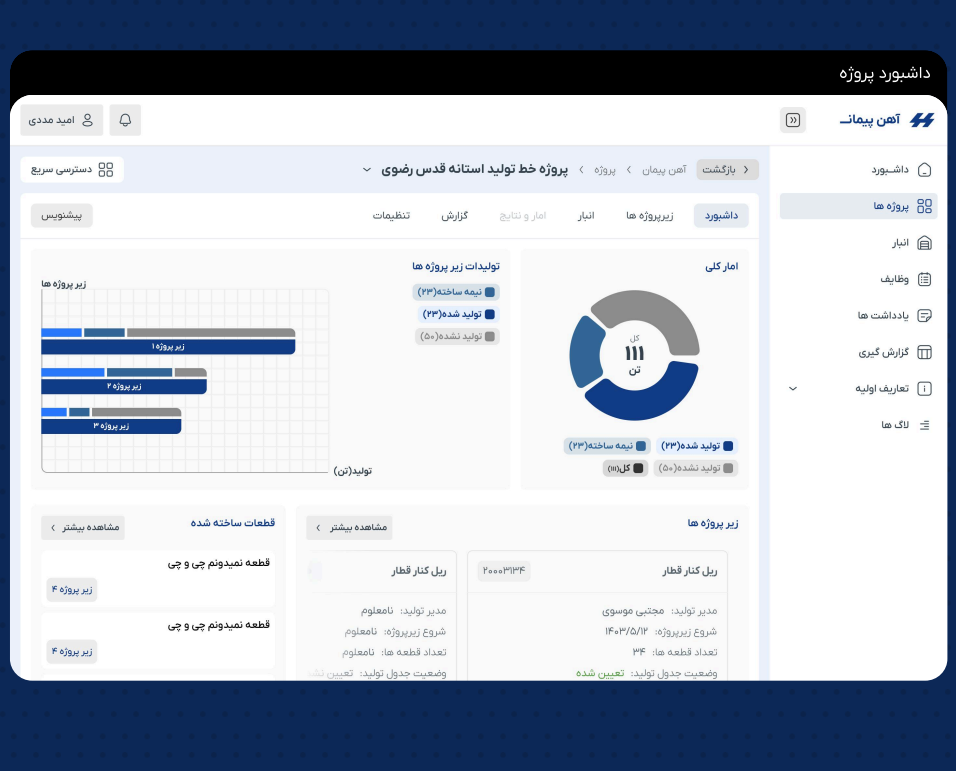
داشبورد پروژه



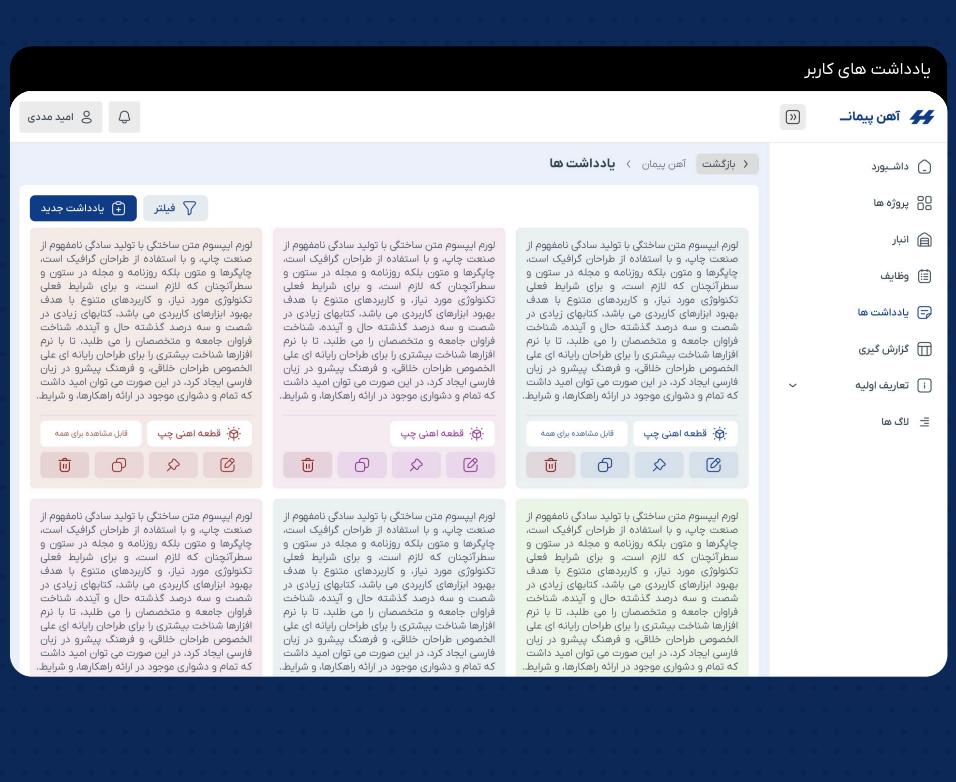
مدیریت کاربران



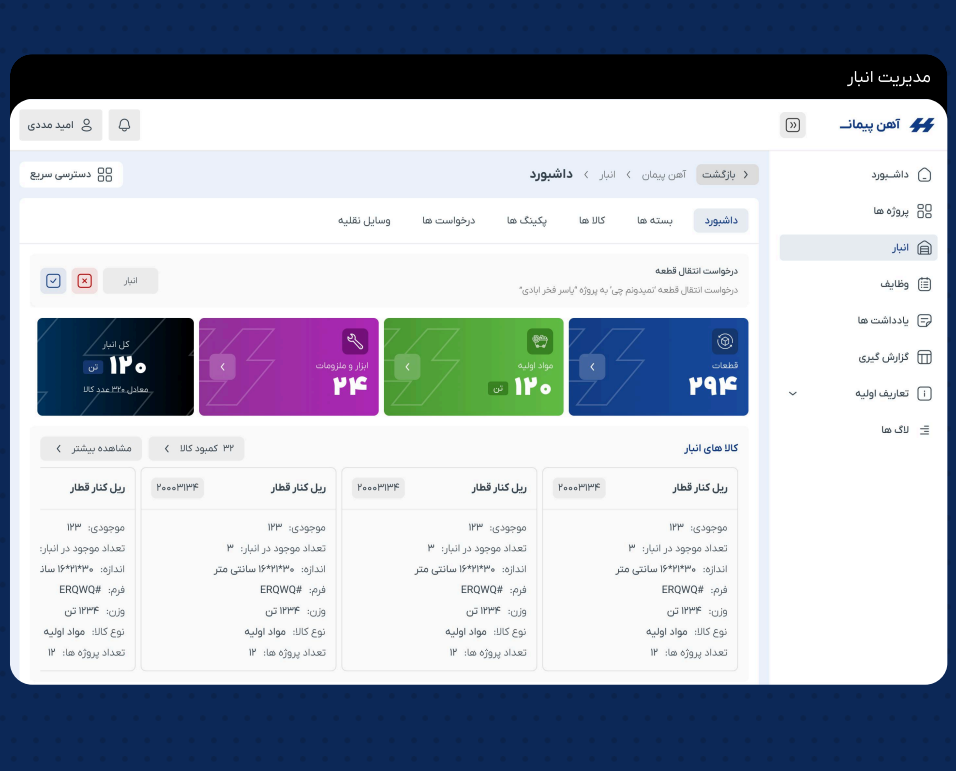
نمایش امار پروژه



داشبورد پروژه



یادداشت های کاربر



مدیریت انبار

DESIGN PRINCIPLES & CONSTRAINTS

تصمیم‌ها، محدودیت‌ها و اعتبار حرفه‌ای کار

وضعیت‌ها فقط برچسب نبودند

در محصولی که هر قطعه میان تولید، QC، انبار و ارسال حرکت می‌کند، وضعیت بخشی از منطق محصول است. Badgeها، دسترسی به اقدام‌ها و پیام‌های تأیید بر اساس وضعیت طراحی شدند تا کاربر هم بداند چه اتفاقی افتاده و هم فقط اقدام‌های معتبر را ببیند.

شفاف‌گویی درباره‌ی محدودیت‌ها

طراحی بر پایه‌ی جلسات با مدیرعامل و رئیس IT، مستند نیازمندی‌ها و مرورهای طراحی شکل گرفت. پروژه پیش از توسعه متوقف شد؛ بنابراین داده‌ای از استفاده‌ی واقعی، تست کاربردپذیری با کاربران نهایی یا KPIهای پس از اجرا وجود ندارد. این محدودیت را به‌جای پرکردن با ادعاهای فرضی، بخشی از روایت پروژه نگه داشته‌ام.

گام اعتبارسنجی بعدی

اگر پروژه ادامه پیدا می‌کرد، مشاهده‌ی میدانی در کارخانه، مصاحبه با اپراتورها و مسئولان QC و انبار، تست کاربردپذیری جریان ثبت تولید، مرور سناریوهای خطا و بررسی فنی فلوها با تیم توسعه، اولویت‌های مرحله‌ی بعد بودند.

OUTCOME & LEARNINGS

خروجی و نتیجه

در پایان دو ماه، معماری اطلاعات، جریان‌های اصلی کاربران، وایرفریم‌ها، رابط‌های نهایی و کامپوننت‌های موردنیاز برای ۱۲ بخش اصلی محصول آماده شد. خروجی طراحی شامل داشبوردهای نقش‌محور، تولید قطعه، کنترل کیفیت، مدیریت پروژه و زیرپروژه، انبار، بسته‌بندی، ارسال، کاربران و سطوح دسترسی، وظایف و سیستم کامپوننت‌ها بود.

پروژه پس از تکمیل طراحی وارد مرحله‌ی توسعه نشد؛ بنابراین داده‌ای از استفاده‌ی واقعی کاربران یا KPIهای پس از اجرا وجود ندارد. به همین دلیل از بیان نتایج فرضی خودداری می‌کنم. خروجی واقعی پروژه، تبدیل یک مجموعه فرایند پیچیده و وابسته به یک ساختار محصول روشن و قابل توسعه بود.

آموخته‌ها و گام بعدی

این پروژه دوباره به من نشان داد که در محصولات سازمانی پیچیده، طراحی رابط نقطه‌ی شروع نیست. ابتدا باید منطق کسب‌وکار، مسئولیت‌ها، ترتیب عملیات و وابستگی تصمیم‌ها را فهمید؛ سپس می‌توان جریانی ساخت که واقعاً قابل استفاده باشد.

اگر پروژه وارد مرحله‌ی بعد می‌شد، پیش از توسعه مشاهده‌ی میدانی فرایند تولید، مصاحبه با اپراتورها و مسئول QC و انبار، تست کاربردپذیری فلو ثبت تولید، بررسی سناریوهای خطا و اعتبارسنجی معماری اطلاعات را انجام می‌دادم.