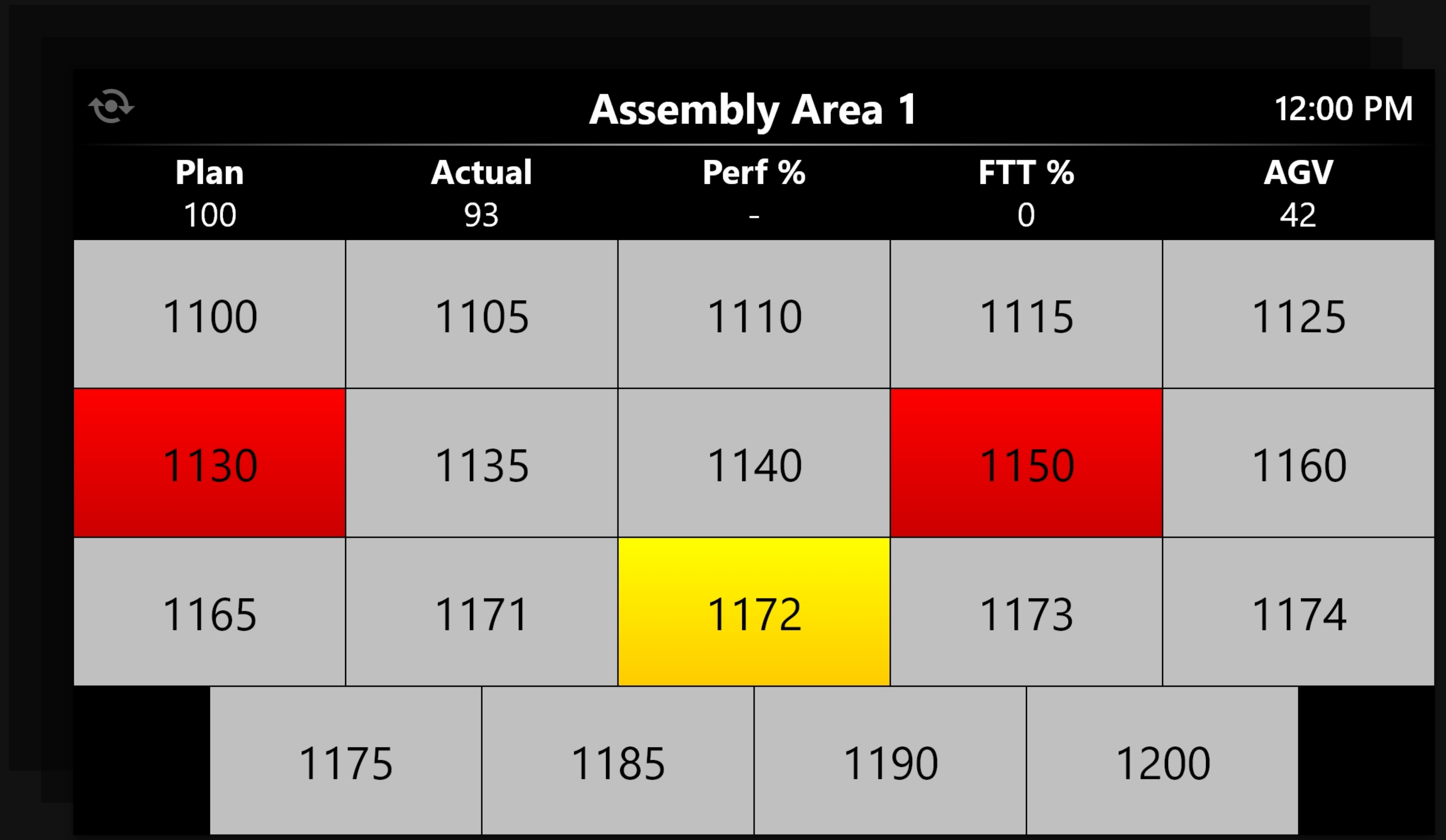


Assembly Area 1 12:00 PM

Plan	Actual	Perf %	FTT %	AGV
100	93	-	0	42

1100	1105	1110	1115	1125
🟥 1130	1135	1140	🟥 1150	1160
1165	1171	🟨 1172	1173	1174
1175	1185	1190	1200	

Case Study

Factory Status Boards

How an admin page redesign helped to fix inaccuracies with live production data and improved usability

Tony Adams

Background

My team had worked for several years with our client, an automotive engine manufacturer in Detroit who standardized several of their manufacturing lines on ProViewDS, a software development platform offered by my company and tailored for industrial control systems.

Over time we co-developed a solution including real-time event logging, visualization of station statuses and production metrics on overhead screens, audio alarms, analytics dashboards, and more.

Project type

Industrial manufacturing
Real-time alarms/events
Data collection
Visualization

Technology

Adobe XD
ProViewDS
C#

My role

UI/UX designer
PM

Problem statement

Our client reported issues where metrics appeared incorrectly on overhead displays, and some values entered on admin pages were not saving after a restart. At the time, my team was also scoping a new project for the client to roll out a similar application to another manufacturing line.

Goals

1. Identify the root cause of inaccurate metrics and apply fixes.
2. Implement usability improvements to admin pages and fix fields that fail to save.
3. Define a standard design for consistency between the new and old apps.

Challenges

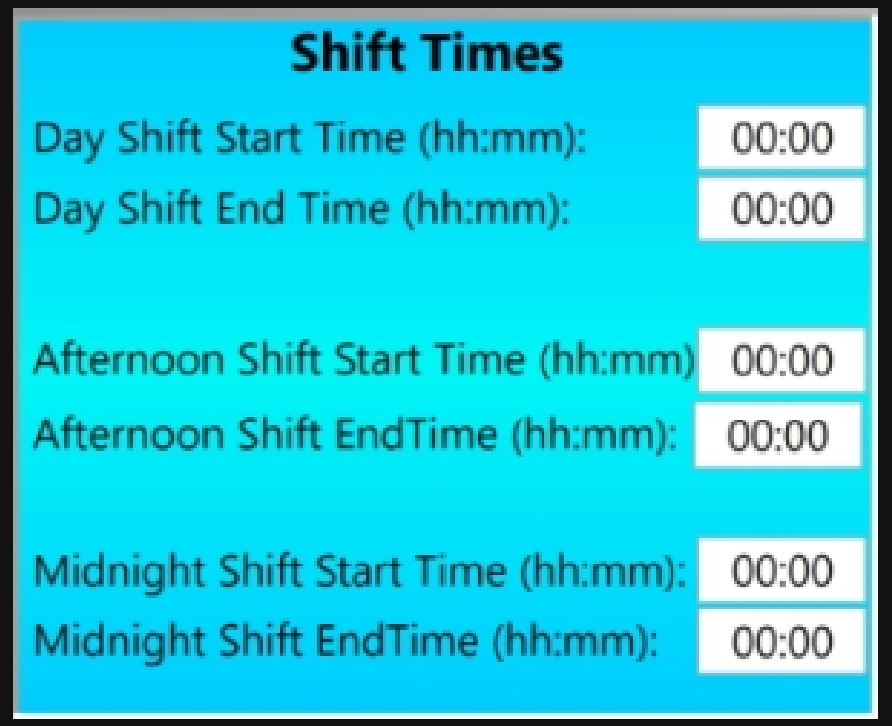
Our client originally developed the application in-house. An analysis for possible “technical debt” and some reverse engineering would be required.

The system includes multiple applications running on separate servers for each manufacturing line with significant differences in their implementations. There was plenty of copy + paste but no shared code base or front-end.

Deployment required a short maintenance window and was “live” with no staging environment or automation. Just back up the files and replace them (always nerve-racking).

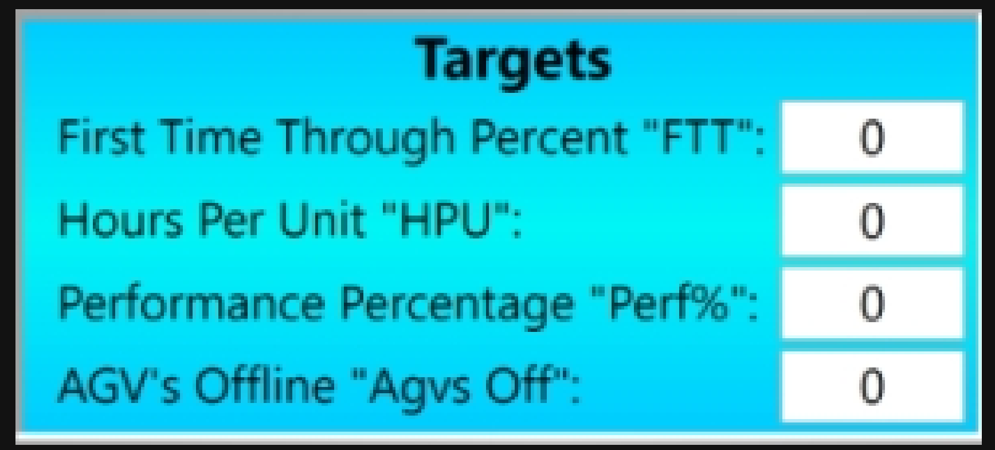
Research

While my developer investigated the cause of any data discrepancies, I worked with my business analyst and our client contact to review the admin pages. After interviewing some users (plant floor managers and team leads), we had a better understanding of the use of the fields and learned that issues usually happened when servers restarted or when schedule adjustments were made.



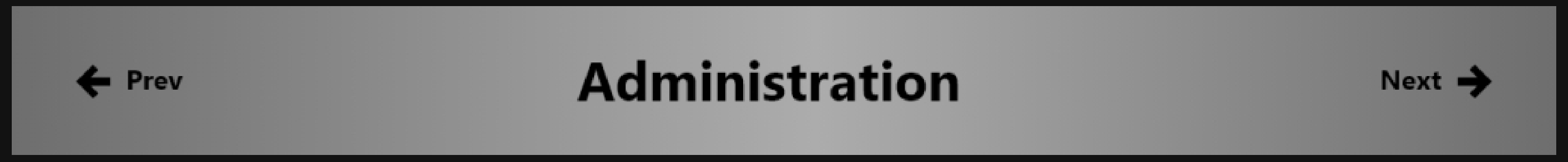
Shift Times

Day Shift Start Time (hh:mm):	00:00
Day Shift End Time (hh:mm):	00:00
Afternoon Shift Start Time (hh:mm)	00:00
Afternoon Shift EndTime (hh:mm):	00:00
Midnight Shift Start Time (hh:mm):	00:00
Midnight Shift EndTime (hh:mm):	00:00



Targets

First Time Through Percent "FTT":	0
Hours Per Unit "HPU":	0
Performance Percentage "Perf%":	0
AGV's Offline "Agvs Off":	0



Shift schedule

Team Leads and managers were responsible for updating the shift schedule daily.

Reviewing a history of our issues with the client, we found inaccurate metrics were reported in the past when someone forgot to update the shift times in the app (on a holiday, half-day, or late start, for example).

We also learned the schedule was stored in a separate system we could easily query rather than requiring users to perform frequent data entry.

Non-retentive tags

Many values in ProViewDS are stored in tags (you can think of them as data fields or variables).

Some tags used on admin pages were not configured to be retentive (a built-in feature to save data when the app shuts down).

This was missed by the original developer, but was an easy fix.

Navigation

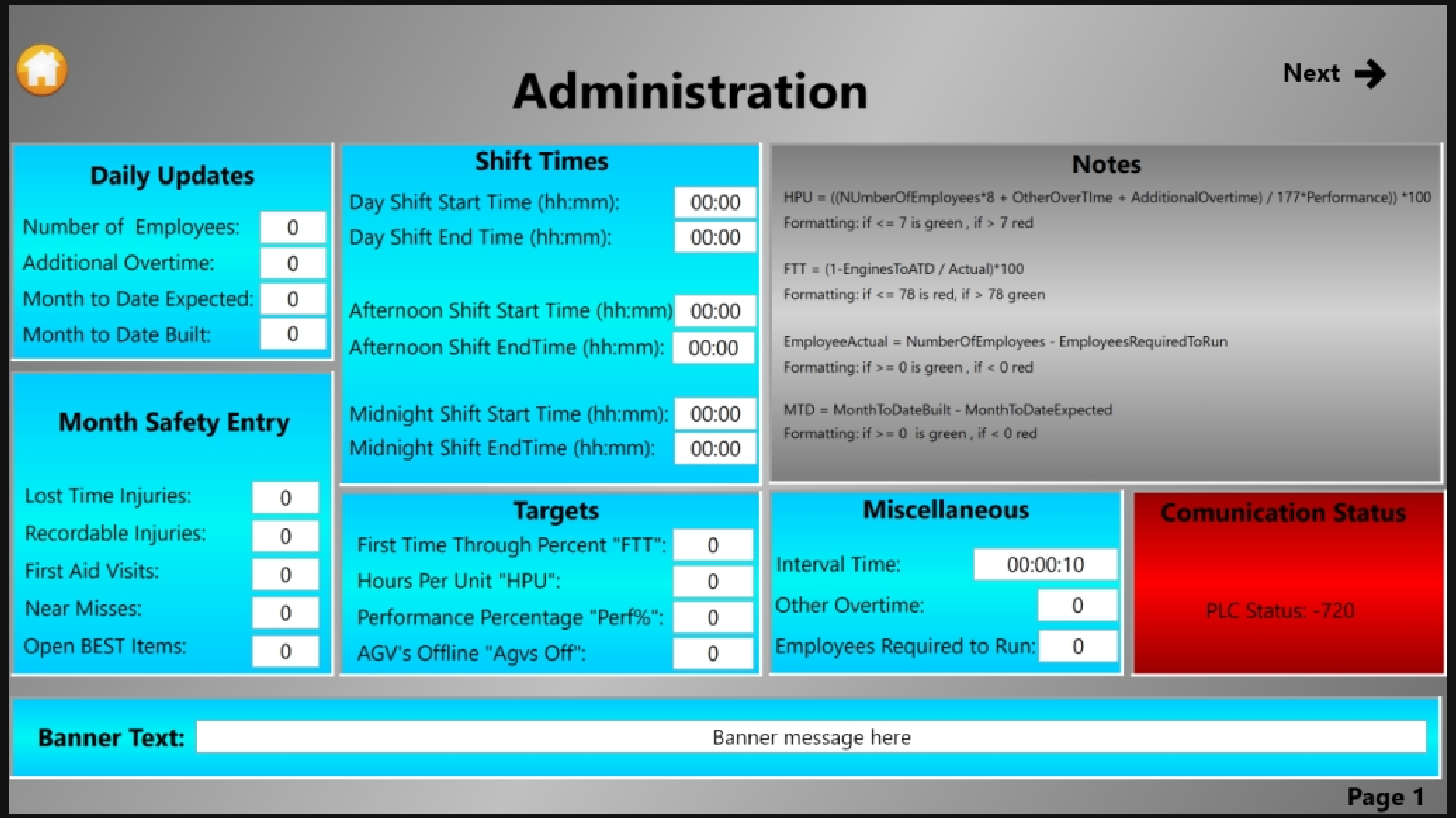
Some users were also frustrated with navigating between pages.

The pages were likely built in piecemeal (starting with one page, adding a second for more real-estate, and so on).

Simple left and right arrow buttons were used to navigate like a slideshow and required users to click through multiple pages to reach pages 3, 4, or 5.

Usability testing

I performed independent usability testing and further dissected the UI implementation in the software.



Next

Administration

Daily Updates

Number of Employees:	0
Additional Overtime:	0
Month to Date Expected:	0
Month to Date Built:	0

Month Safety Entry

Lost Time Injuries:	0
Recordable Injuries:	0
First Aid Visits:	0
Near Misses:	0
Open BEST Items:	0

Shift Times

Day Shift Start Time (hh:mm):	00:00
Day Shift End Time (hh:mm):	00:00
Afternoon Shift Start Time (hh:mm)	00:00
Afternoon Shift EndTime (hh:mm):	00:00
Midnight Shift Start Time (hh:mm):	00:00
Midnight Shift EndTime (hh:mm):	00:00

Notes

HPU = ((NUmberOfEmployees*8 + OtherOverTIme + AdditionalOvertime) / 177*Performance)) *100
Formatting: if <= 7 is green , if > 7 red

FTT = (1-EnginesToATD / Actual)*100
Formatting: if <= 78 is red, if > 78 green

EmployeeActual = NumberOfEmployees - EmployeesRequiredToRun
Formatting: if >= 0 is green , if < 0 red

MTD = MonthToDateBuilt - MonthToDateExpected
Formatting: if >= 0 is green , if < 0 red

Targets

First Time Through Percent "FTT":	0
Hours Per Unit "HPU":	0
Performance Percentage "Perf%":	0
AGV's Offline "Agvs Off":	0

Miscellaneous

Interval Time:	00:00:10
Other Overtime:	0
Employees Required to Run:	0

Comunication Status

PLC Status: -720

Banner Text: Banner message here

Page 1

Old admin page

Fields saved immediately

The admin pages instantly saved values when users hit Enter, clicked, or tabbed away from any field. This incurred some usability issues:

1. If more than 1 field contributed to the same metric, the system showed an incorrect number until the user updated each field. This caused skewed data to appear on overhead screens for minutes at a time. Avoidable if all values are updated simultaneously with 1 click.

2. If a user entered invalid values in some formatted fields, it would reset to 0 rather than returning to the original value. If the system was running during a shift, this could instantly cause production metrics to appear off until a valid number is entered, besides frustrating the user.

Enabled/disabled fields

Some fields were disabled but gave the user no visual indication. Other fields appeared as plain text but were editable. Though these fields were not wired up to save entered data, they were confusing to users who'd happen to interact with them.

Screen resolution

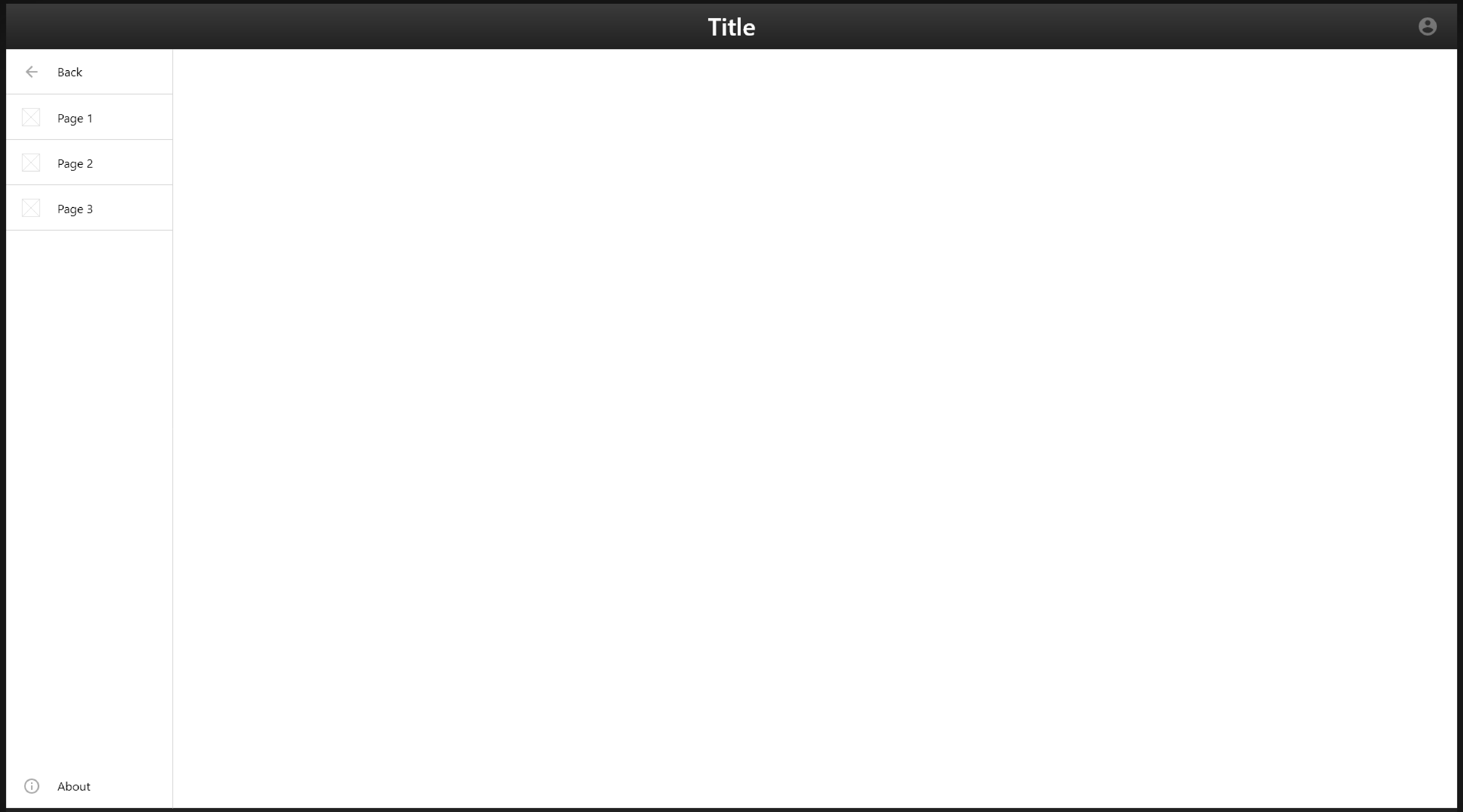
The layout was hardcoded to render in 1366 x 768 resolution, but it was used on standard 1920 x 1080 screens by most users.

Look and feel

The visual design appeared severely outdated, along with sizing and layout issues across several pages. I suggested that we update the design to our latest standards and organize the content.

Design

I had previously built a design system and UI kit for my team's ProViewDS projects, so standard UI controls and the new layout were easy to import. After creating a look and feel mock-up for our client's review, I built template pages and side navigation for developers to re-use. The platform uses a non-responsive front-end, so sizing and spacing needed to be pixel perfect. By setting the new standard and building more easily reusable components, we planned to update the new project and then copy to the other existing applications.



New admin layout

Solution

1. Metrics fixes

My developer had quickly identified other possible causes of inaccurate metrics. The application was not designed to recalculate metrics on the fly. Instead, it relied on a shaky system of incremental counters and assumed the application would only ever restart between shifts (which was not always the case). We hit the whiteboard, laid out a new workflow accounting for edge cases, and proposed the changes to our client. They also corrected retentive tags to save data properly, even when the application was restarted (during a server reboot, for example). This fixed the more immediate issues experienced by users in the existing application.

2. Admin pages improvements

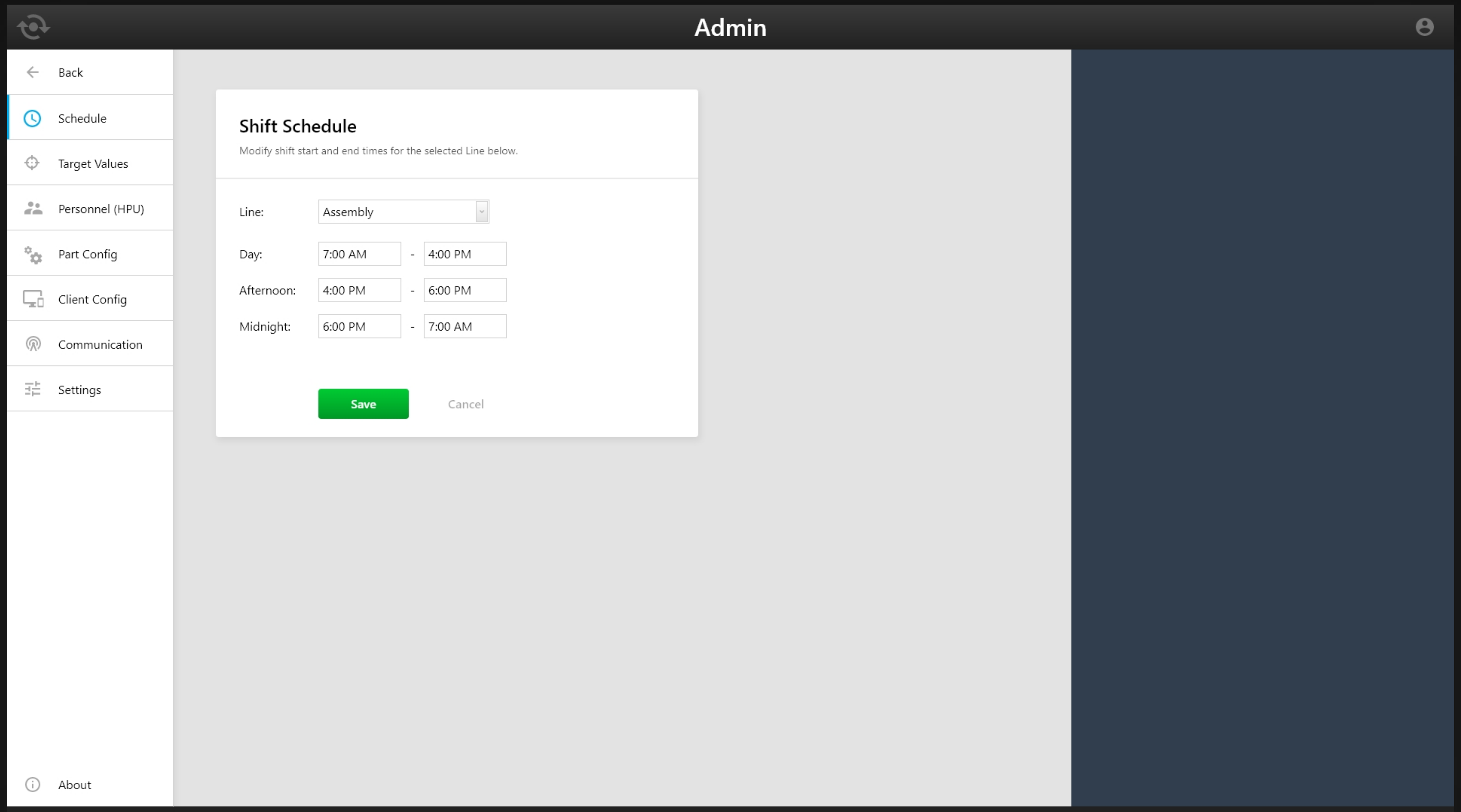
Using my templates and UI kit, we implemented admin pages in the new application including proper screen resolution, more user-friendly and organized navigation, improved the layout and styling of the content.

Save and Cancel buttons were implemented where appropriate, allowing users to change multiple values before saving and avoiding incorrect values on overhead displays.

The shift schedule page was updated to pull and save to a centralized database so separate applications running on multiple lines could share the data and allow users to update the schedule from any of the applications.

3. Standardization

The new design standards for the admin pages and the “on the fly” method of calculating metrics were implemented to each manufacturing line and copied to other new applications my team built for our client. We received very positive feedback as the improvements rolled out over time.



Shift Schedule

Modify shift start and end times for the selected Line below.

Line: Assembly

Day: 7:00 AM - 4:00 PM

Afternoon: 4:00 PM - 6:00 PM

Midnight: 6:00 PM - 7:00 AM

New shift schedule page