

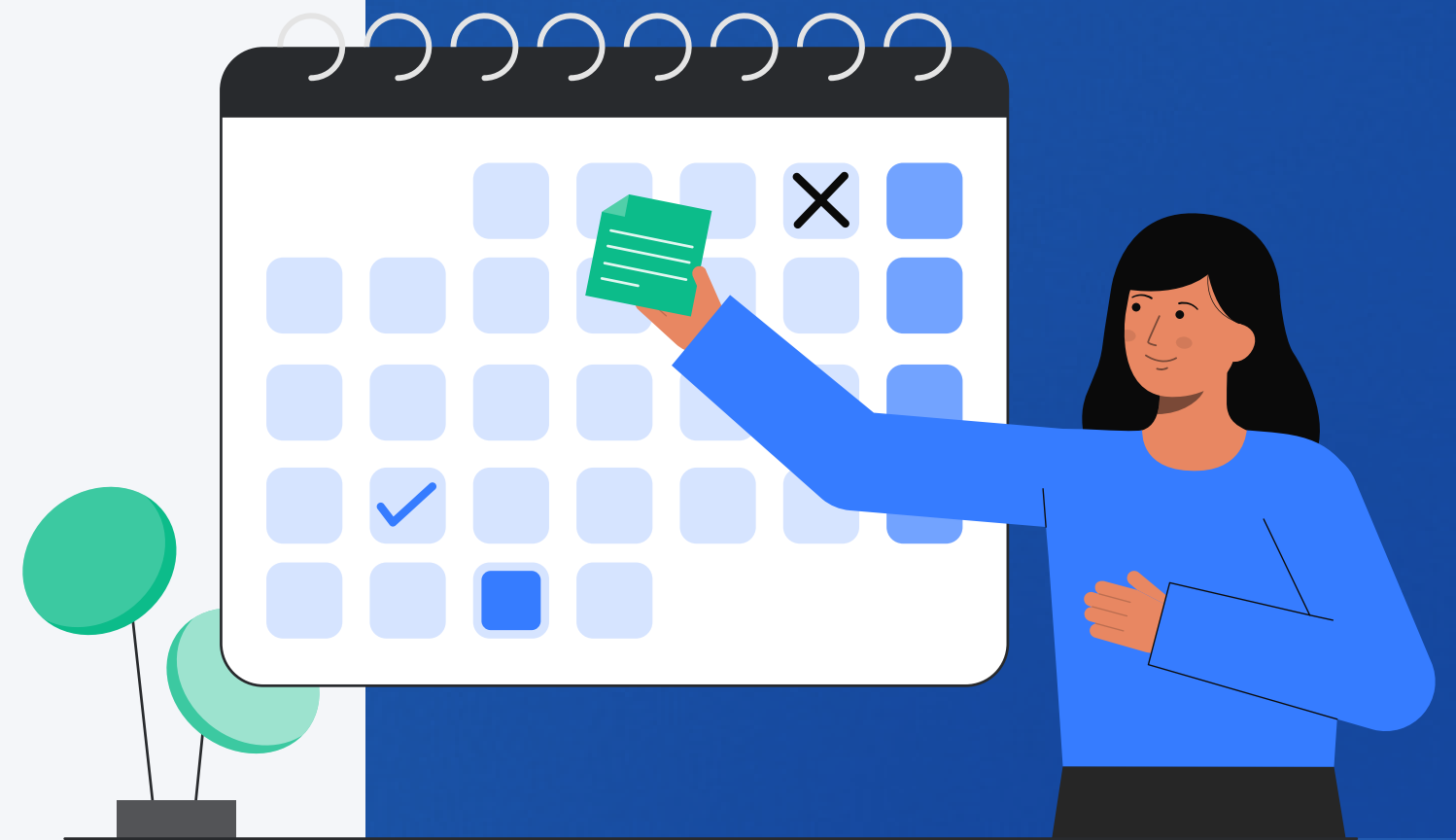
Printo Kallukaran

Design with logic & imagination

Designing a smart nurse shift roster  
system for high-complexity  
healthcare operations

UX

Case study



# Problem statement

An application to create rosters for nurses working on different shifts considering the following constraints.

- ✓ A nurse does not work the day shift, night shift and late night shift on the same day (i.e. no 24-hour duties).
- ✓ A nurse may go on a holiday and will not work shifts during this time.
- ✓ A nurse does not do a late night shift followed by a day shift the next day.
- ✓ Two nurses dislike each other and thus cannot work on the same shift because of that.
- ✓ One nurse is lazy and must be paired with hard worker.
- ✓ A shift required a charge nurse.



# Understand [Breakdown the problem statement]

- ✓ A nurse does not work the day shift, night shift and late night shift on the same day (i.e. no 24-hour duties).
- ✓ A nurse does not do a late night shift followed by a day shift the next day.

- No nurse can work day + night + late-night in the same day.

- No 24-hour duty under any circumstances.

Fatigue (Human constraints)

Shift fairness (Human constraints)

- ✓ A nurse may go on a holiday and will not work shifts during this time.

- Some nurses have leave periods.

- ✓ Two nurses dislike each other and thus cannot work on the same shift because of that.

- Some nurses cannot work together.

Team harmony (Human constraints)

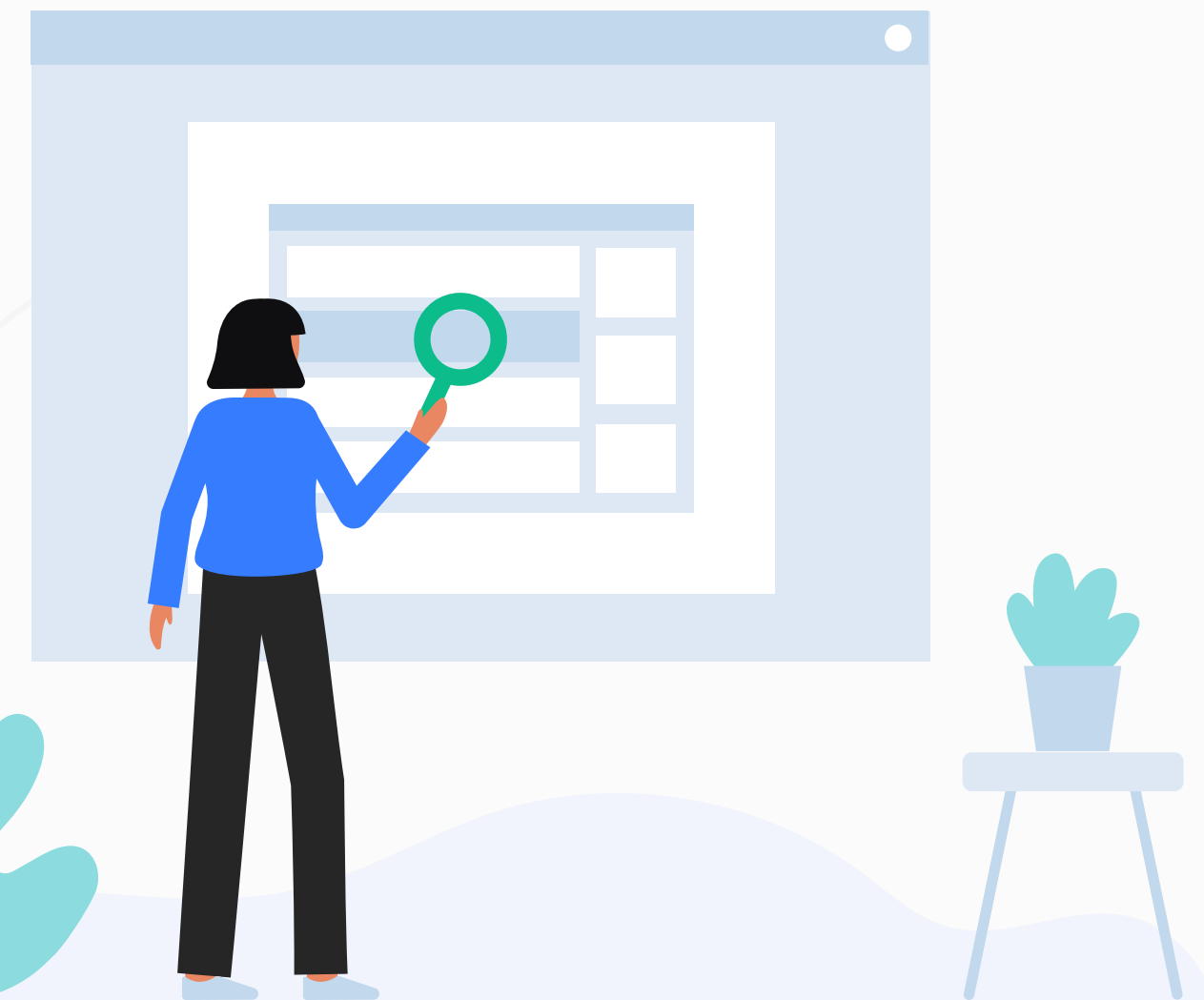
- ✓ One nurse is lazy and must be paired with hard worker.

- One nurse is not dependable (lazy) and must be paired with a high performer.

Skill distribution (Human constraints)

- ✓ A shift required a charge nurse.

- Every shift requires one charge nurse.



# Persona [Assumption based]

## John - Nurse Manager



### Goals

- > Create rosters quickly
- > Avoid complaints
- > Ensure compliant coverage
- > Reduce conflict resolution

### Pain Points

- > Manual tracking of leave & shift limits
- > Interpersonal staff conflicts
- > Lack of visibility on skill/charge nurse distribution

## Nancy - Senior/Charge Nurse



### Goals

- > Maintain team workflow
- > Avoid burnout
- > Ensure the staff member is not overworked.

### Pain Points

- > Being assigned too many leadership-required shifts
- > Late-night to day-shift fatigue

## Riya - Junior Nurse



### Goals

- > Understand schedule clearly
- > Get fair distribution of shifts
- > Avoid working with specific incompatible coworkers

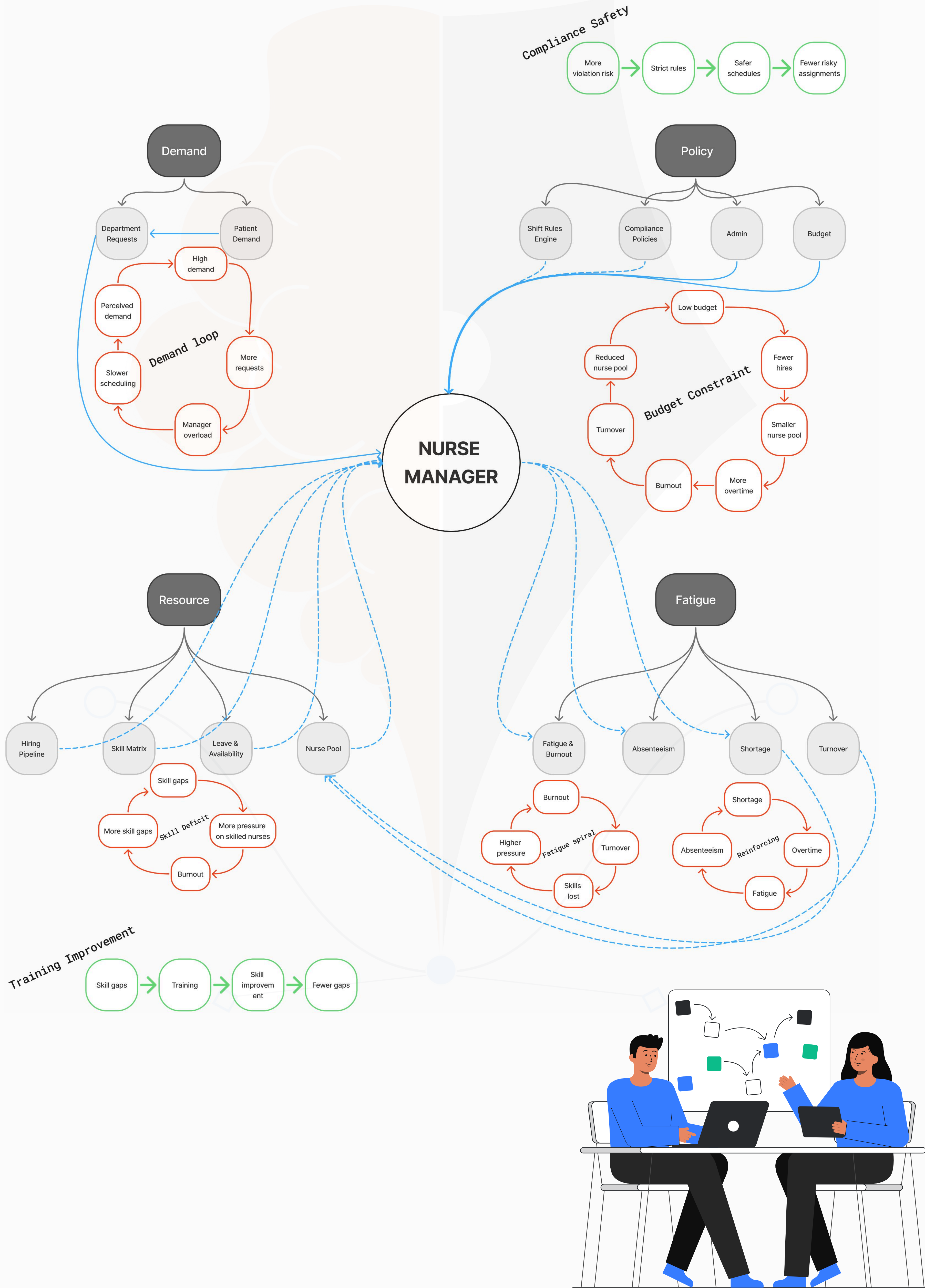
### Pain Points

- > Manual tracking of leave & shift limits
- > Interpersonal staff conflicts
- > No visibility into skill/charge nurse distribution



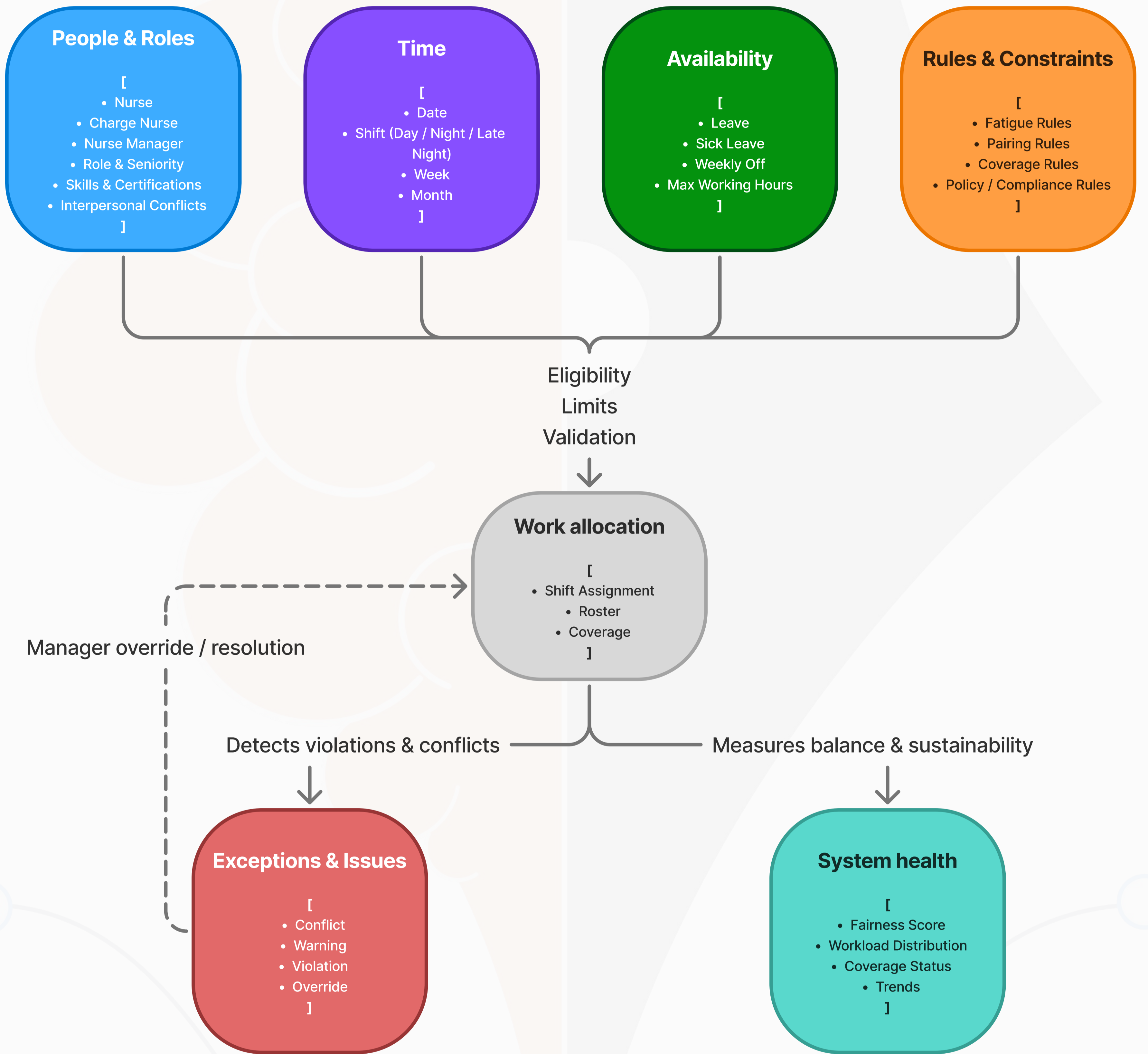
# System Mapping

A system-level view of how demand, rules, resources, and fatigue interact to shape nurse scheduling and where interventions can break recurring issues.



# Constraint-Based work allocation framework

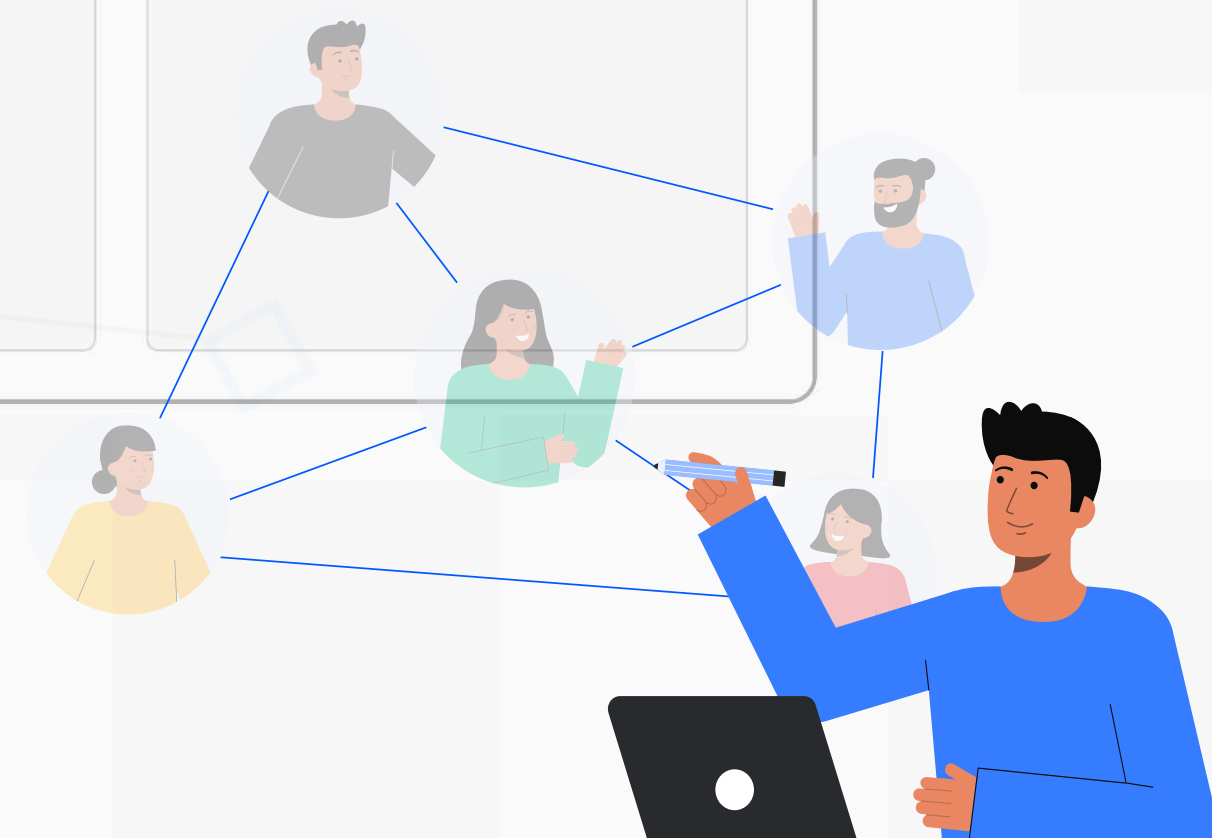
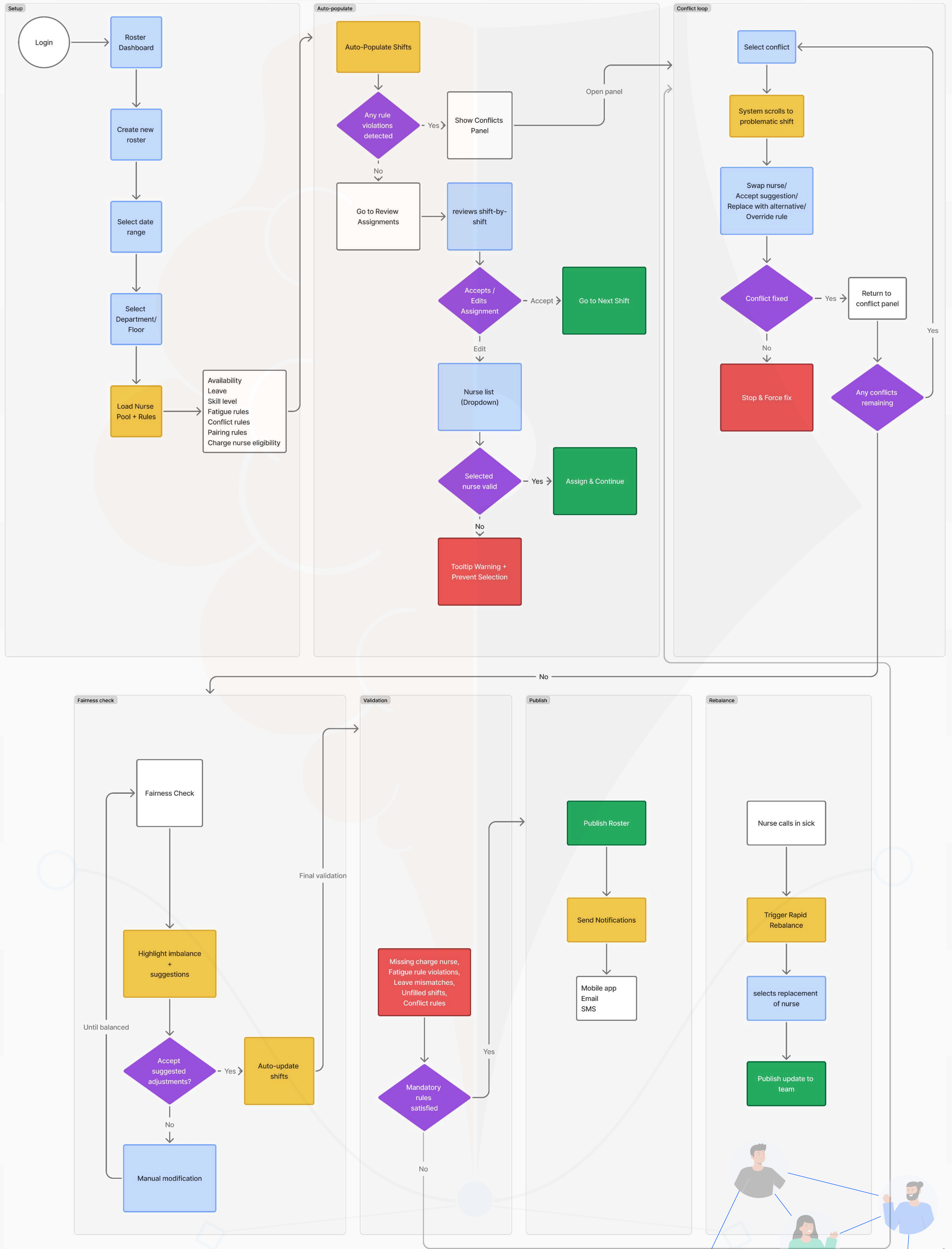
This diagram shows how the system decides who gets assigned to what work, based on availability, eligibility, and rules, with built-in checks for conflicts and balance.



# End-to-End Roster Management user flow

This diagram shows the end-to-end process of creating and managing a nurse roster, from auto-generating shifts based on availability and rules to resolving conflicts, validating fairness, publishing, and rapidly rebalancing schedules when real-world changes occur.

● User Actions   
 ● System Action   
 ● Decision Nodes   
 ● Error / Conflict   
 ● Success / Completion States



# Insights / Pain Points Identified

Walking through the user flow revealed hidden friction points that shaped the key product decisions.

## High Cognitive Load

- > Leave data
- > Conflict rules
- > Charge nurse requirement
- > Fatigue constraints
- > Personality conflicts

## High Error Probability

- > Pair incompatible nurses
- > Violate late-night to day rule
- > Forget the lazy nurse pairing constraint
- > Miss charge nurse assignment

## Repetitive Manual Work

- > Every shift demands the same steps:  
Assign > Validate > Fix > Move

## Conflicts Are Hard to Notice

- > Errors appear after the assignment, not during it.
- > This interrupts flow and creates manual rework.

## Fairness Cannot Be Evaluated During the Flow

- > Overused staff
- > Underutilized staff
- > Charge nurse distribution

## Human Constraints Are Invisible to the System

- > The roster tool needs to enforce constraints proactively, not reactively.



# Key UX Decisions

Each key decision arose directly from real scheduling challenges, shaping features that make the system smarter, safer, and effortlessly usable.

## Auto-assignment with a 'Constraint Engine'

- > Riya cannot work with Ananya
- > Lazy nurse must be paired with hard worker
- > Charge nurse missing in Night Shift"
- > Late-night shift assigned - next day blocked automatically

## Fairness Meter

- > Overworked nurses
- > Under-utilized nurses
- > Balance score
- > Fatigue constraints
- > Personality conflicts

## Smart conflict timeline

- > If someone is on leave-  
Their name is visually "faded + locked"
- > If a nurse is incompatible with someone already placed -  
Their name shows a 'discreet' tag
- > If a fatigue rule is violated-  
System forces confirmation (manual override allowed but logged)

## Pairing Automation

- > If lazy nurse is selected-  
System auto-selects one of the available hard workers.  
If none are available > conflict flag



# Experience Principles

To translate insights into a coherent experience, I defined guiding principles that anchor the design and shape how the system should behave.

## Reduce Cognitive Load

- The system should surface complexity only when needed, keeping high-pressure tasks simple.

## Make Constraints Visible, Not Overwhelming

- Rules must appear at the moment of decision, not clutter the UI.

## Prevent Errors Before They Happen

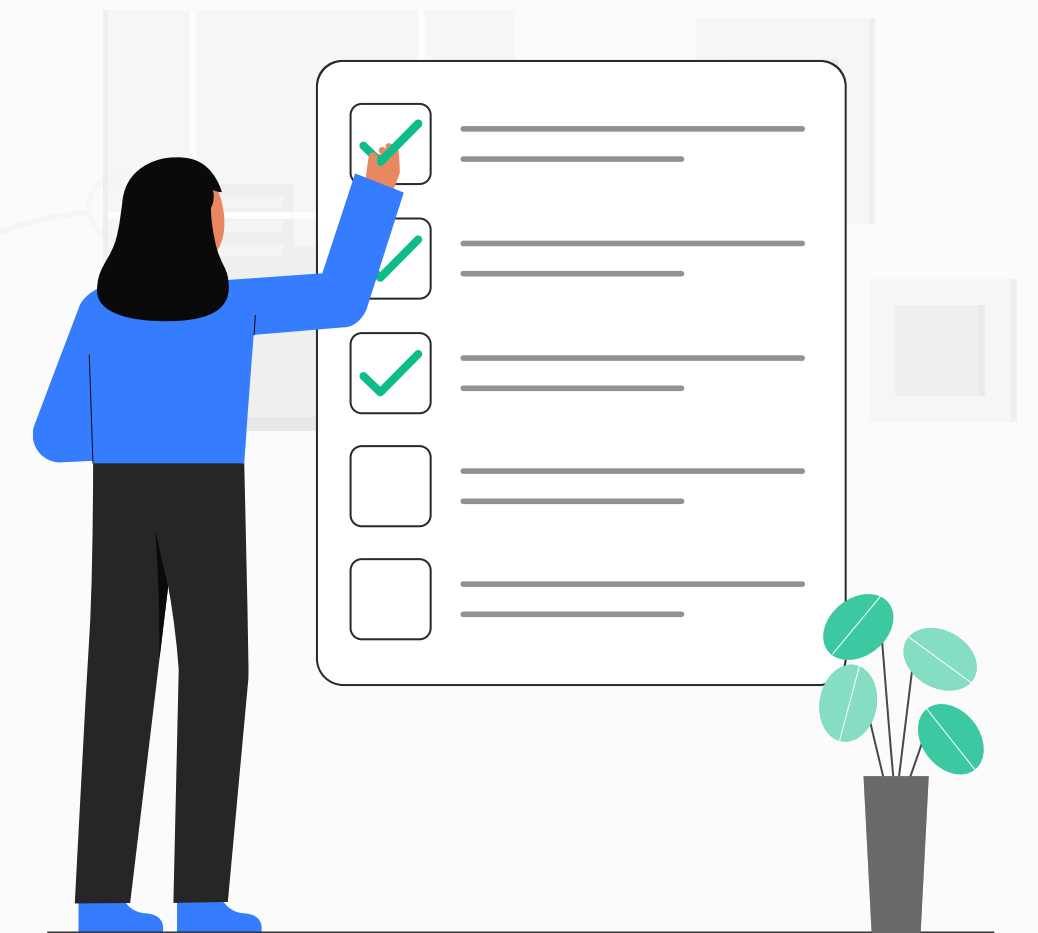
- The interface actively guides managers away from violations instead of fixing issues later.

## Maintain Fairness & Transparency

- Staff distribution must be clear, helping prevent bias or overuse of certain nurses.

## Support Fast, Confident Decisions

- Managers should spend less time diagnosing problems and more time making informed assignments.



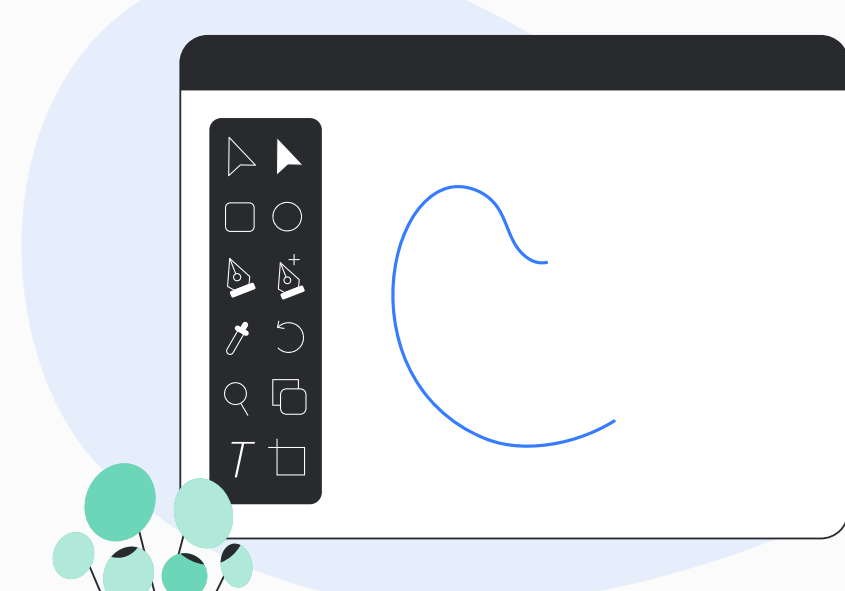
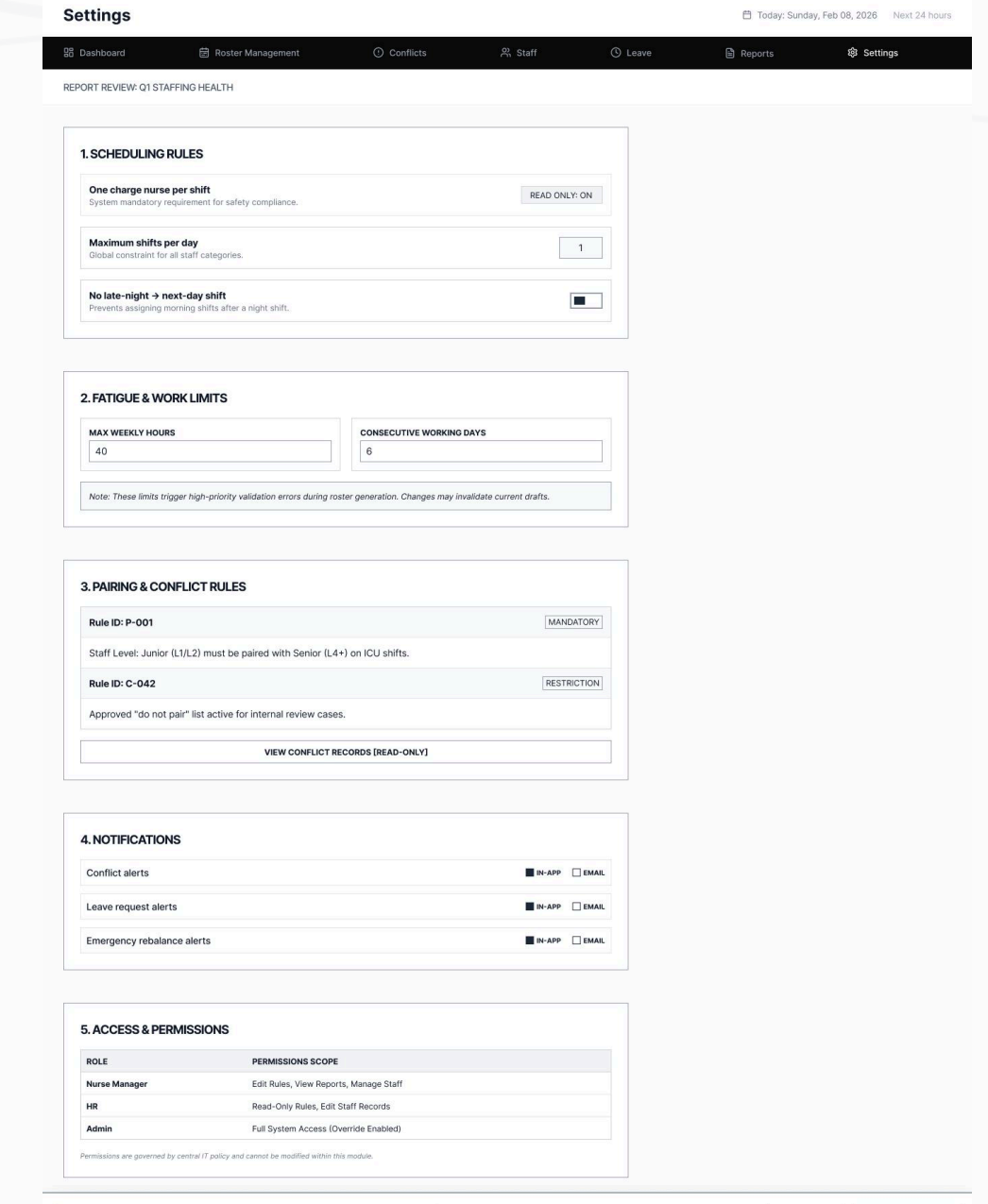
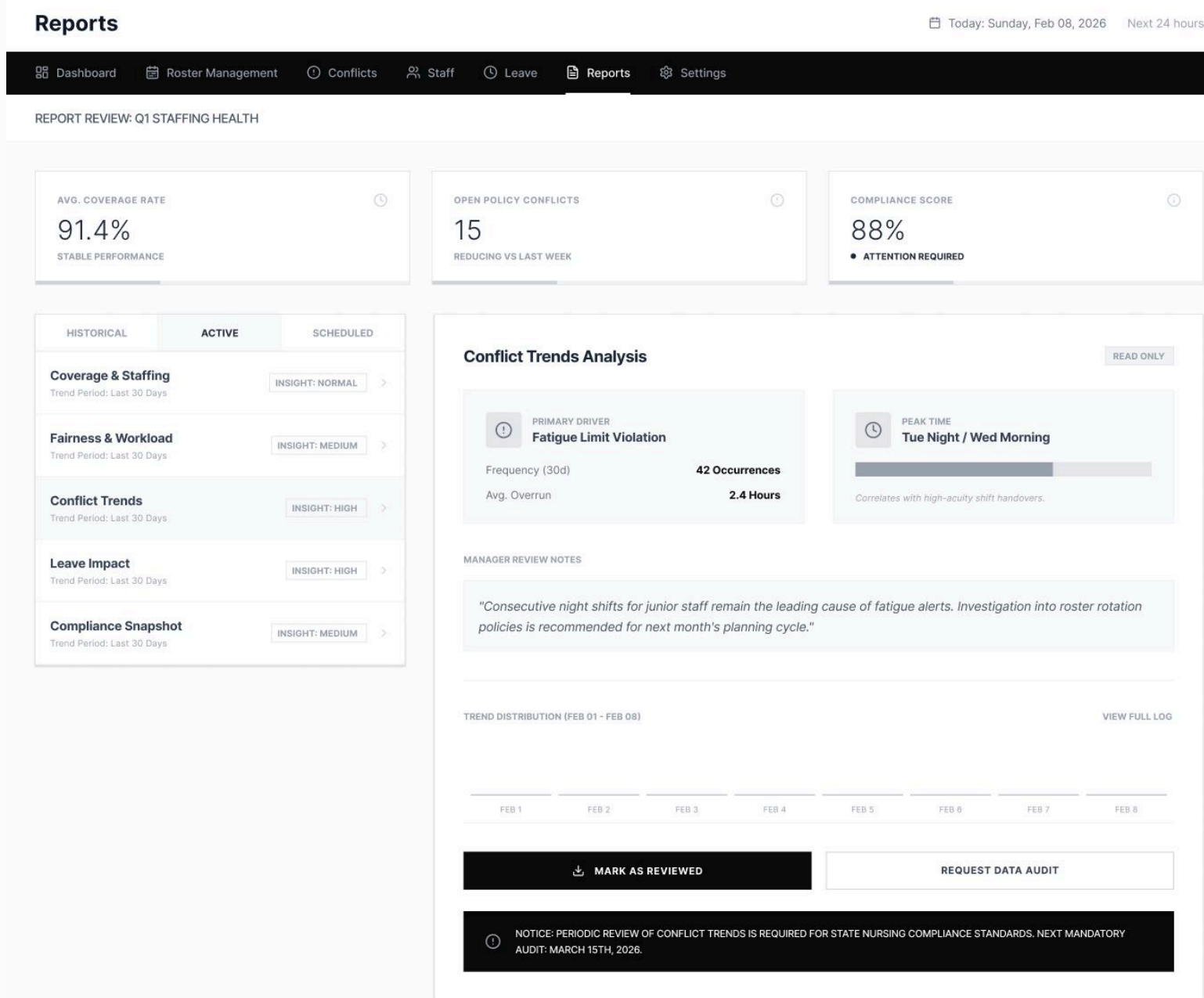
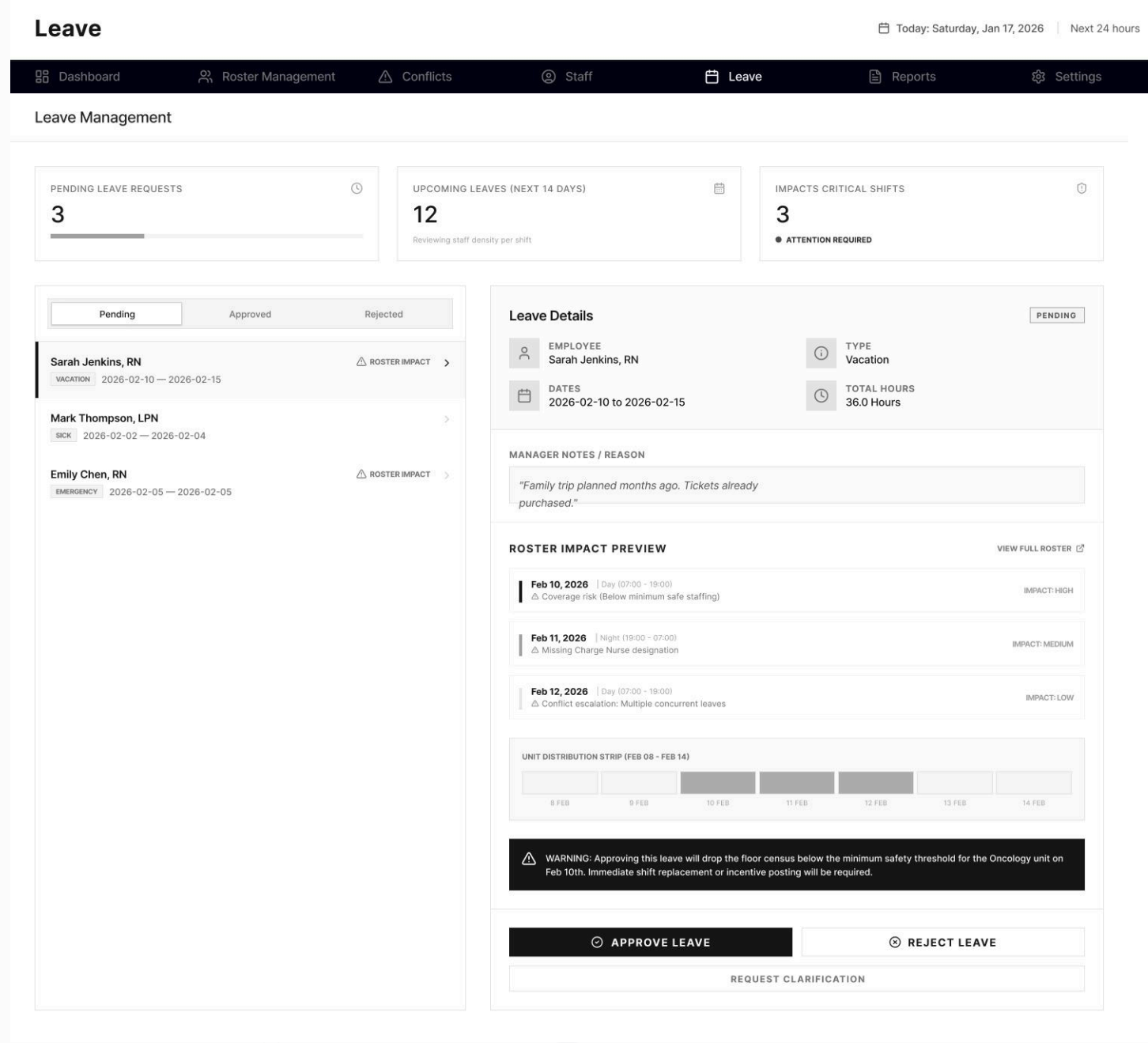
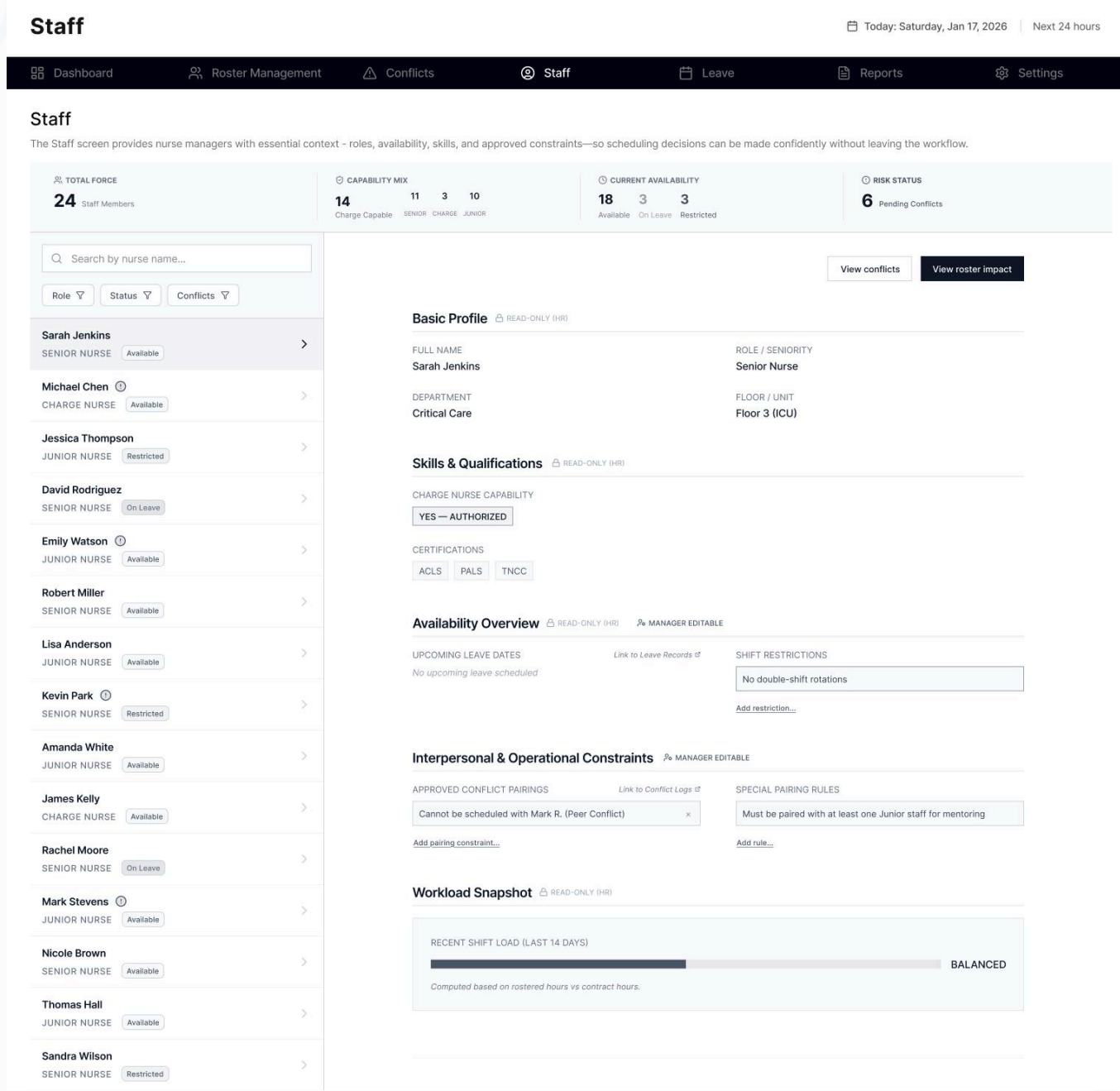
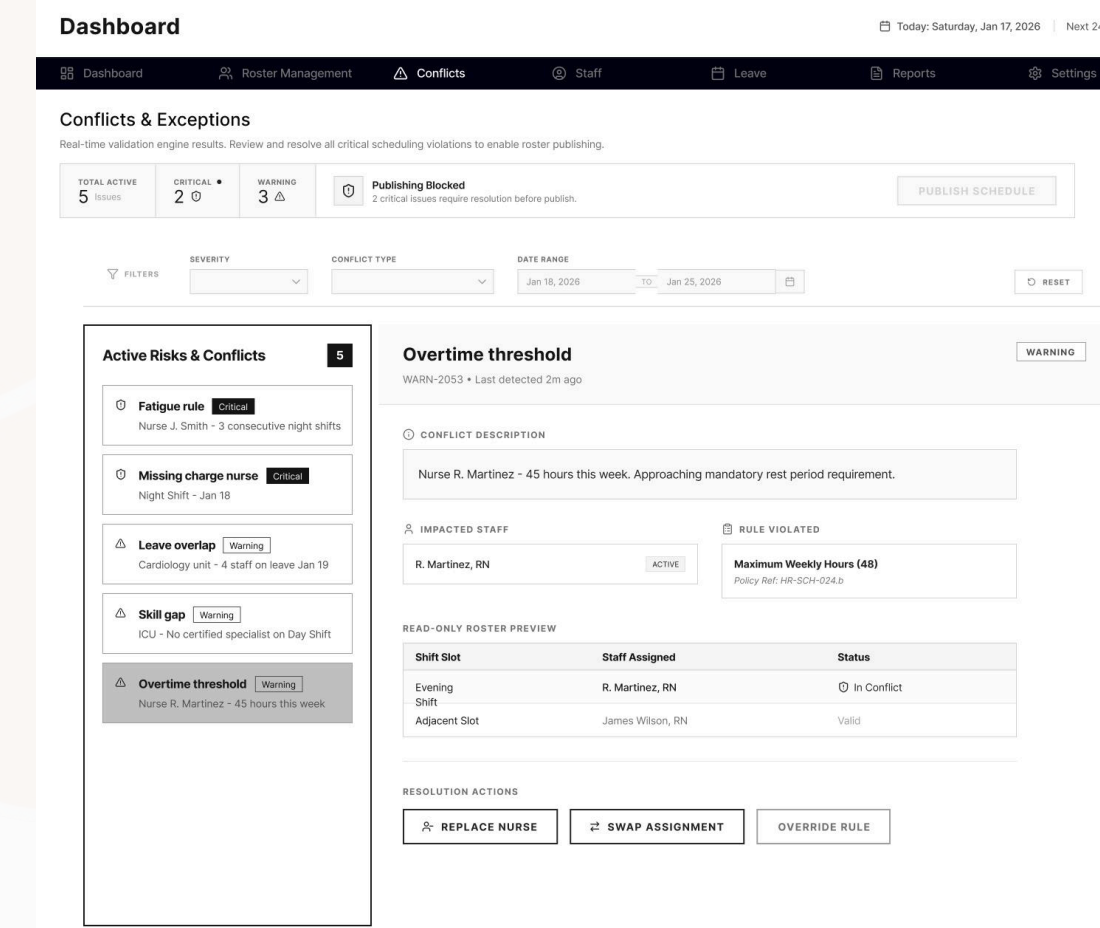
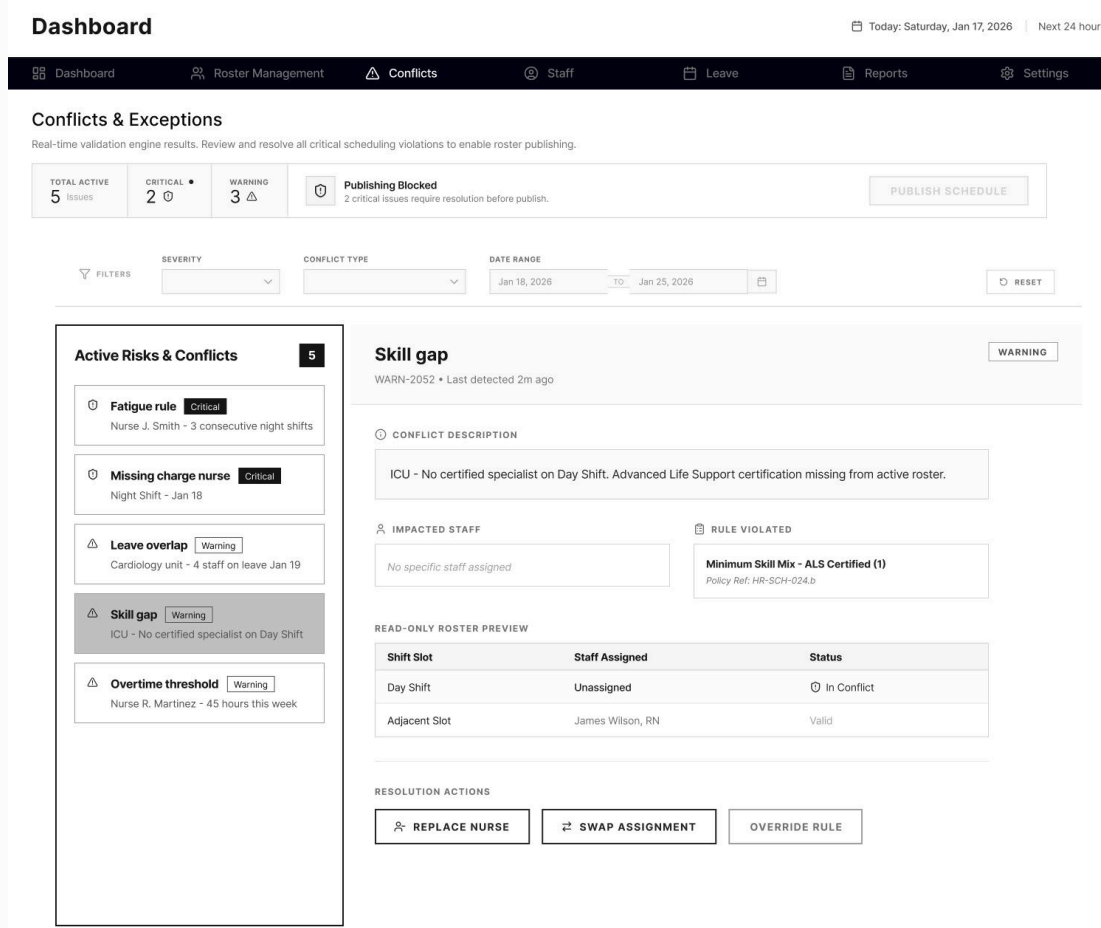
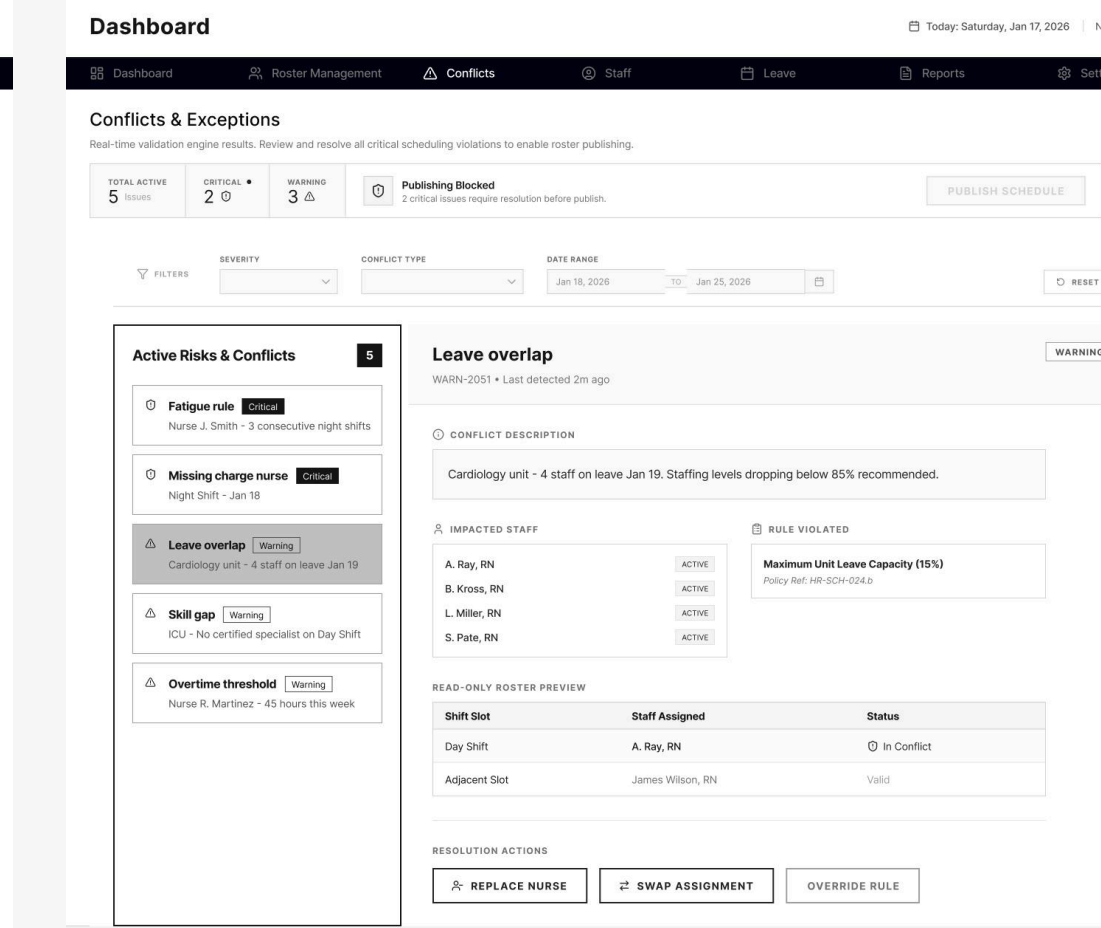
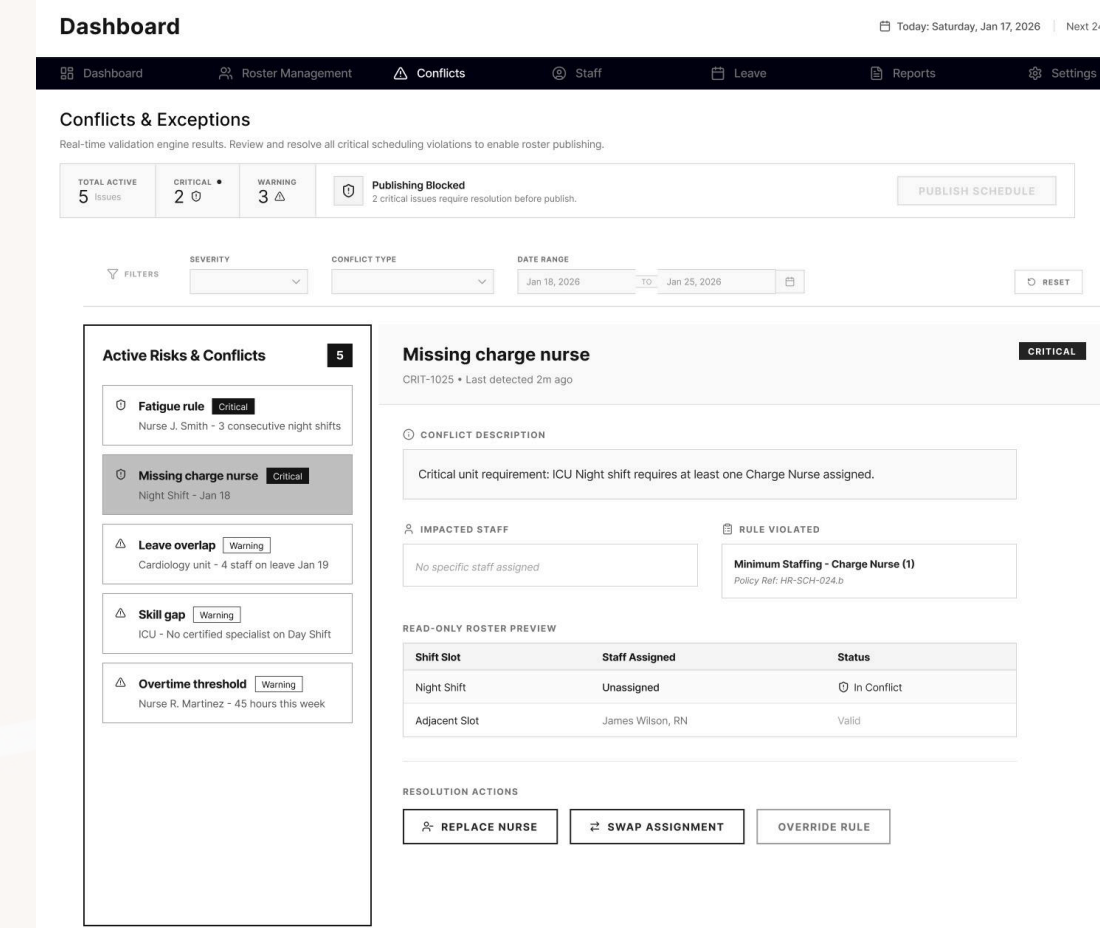
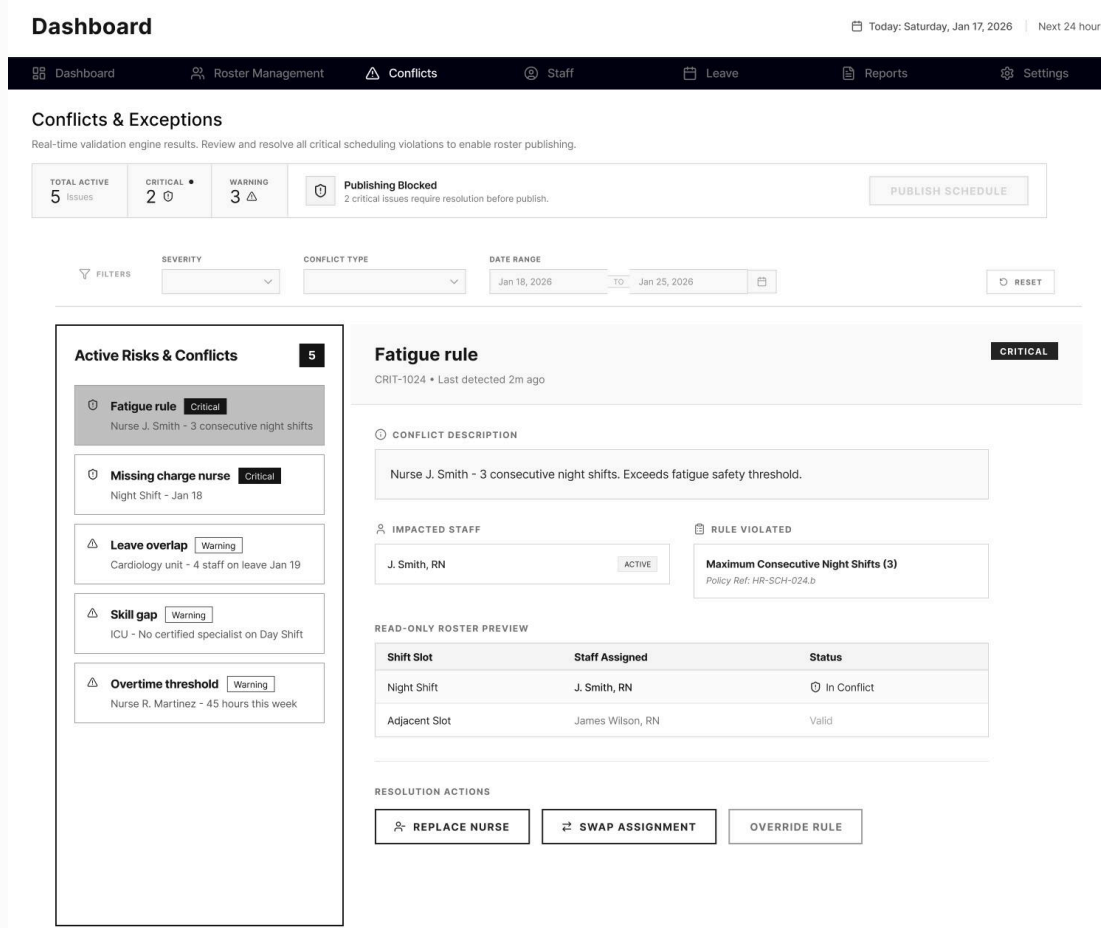
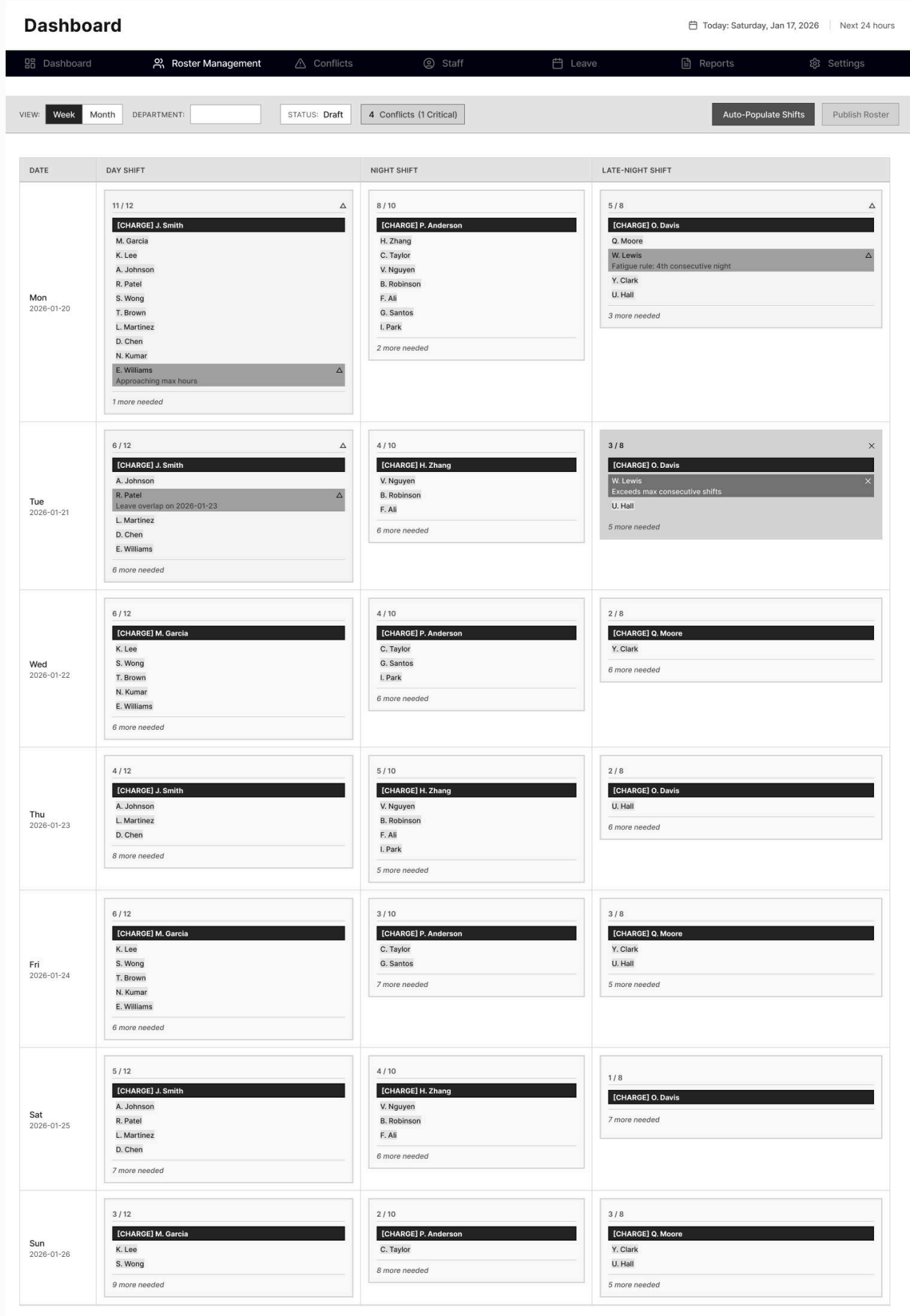
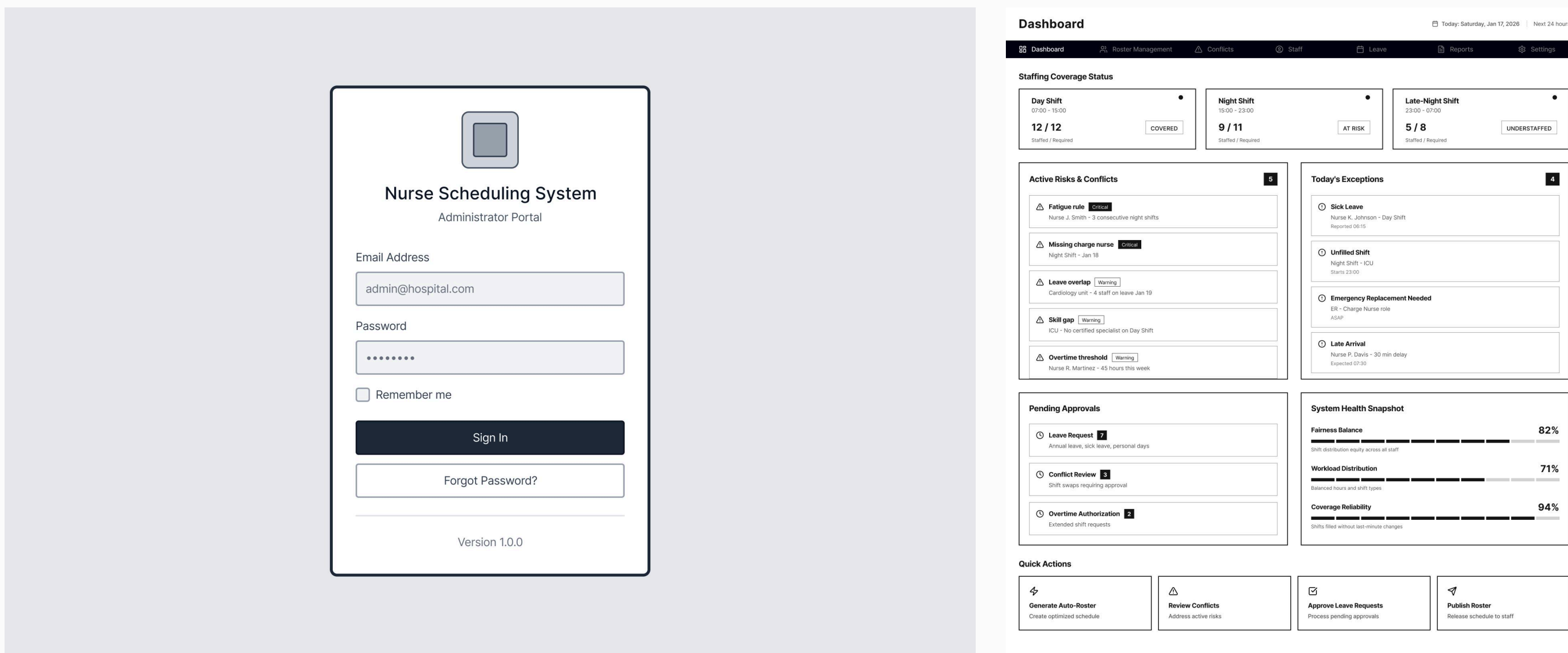
# Feature Priority

Features were prioritized by task criticality and risk. Core and safety features formed the foundation, with enhancements layered progressively to support confident decisions without cognitive overload.

Feature type	Feature	Why it's require	If removed
● Core	Shift calendar / Grid view	Central workspace to assign and review shifts across days and shift types	Roster creation collapses
● Core	Nurse assignment per shift	Enables direct staffing decisions	No roster can be created
● Core	Availability & Leave filter	Prevents assigning unavailable nurses	High error rate, manual rework
● Core	Charge nurse	Ensures every shift has leadership coverage	Compliance & safety risk
● Core	Publish Roster	Finalizes and communicates schedules	System has no outcome
● Safety	Constraint Engine	Validates rules (fatigue, conflicts, leave, shift sequence)	Illegal & unsafe schedules
● Safety	Conflict indicators (Red / Yellow / Green)	Makes violations visible at decision time	Errors discovered too late
● Safety	Late - Night - Day shift block	Prevents nurse fatigue	Patient safety risk
● Safety	Incompatible nurse detection	Avoids known interpersonal conflicts	Team friction, HR escalations
● Safety	Lazy - Hard worker pairing rule	Maintains shift performance balance	Drop in care quality



# Wireframes



# Thanks for your time!

This work focuses on system design, workflows, and interaction clarity; visual design was intentionally out of scope.

