75Guard - Ultimate UI/UX Design Document (Antigravy-Ready)

DESIGN PHILOSOPHY FOUNDATION

What 75Guard Is NOT

❌ An attendance calculator
❌ A student ERP dashboard
❌ A browsing interface
❌ A percentage tracker

What 75Guard IS

✅ **A situational decision-support instrument**
✅ Helps students plan attendance under irreversible academic constraints
✅ **Primary outcome:** User instantly understands where they stand, what they can risk, and what they must not miss

1. CORE UX PHILOSOPHY: STATE-DRIVEN INTERFACE

Fundamental Principle

The UI has NO neutral home screen.
The entire interface adapts based on one of three global states, derived automatically from attendance data.

State-Driven, NOT Screen-Driven

```
Traditional App:       75Guard:

Home → Subjects          [Calculates State] → Renders Appropriate UI

 → Details                ↓

 → Analytics           State dictates:

                     - Layout density

                     - Animation speed

                     - Interaction requirements

                     - Language tone

                     - Visual tension
```

2. THE THREE GLOBAL STATES (PRIMARY DIFFERENTIATOR) 🟢 STATE 1: SAFE Condition:

Attendance ≥ required trajectory (>75%) Recovery margin exists Buffer ≥ 5 classes UX Behavior:

Layout: Relaxed spacing (24px gaps), generous padding Animation: Smooth, slow (600ms transitions), soft easing (cubic-bezier(0.4, 0, 0.2, 1)) Interaction: Exploratory allowed, no confirmations required Visual Feel: Calm, open, breathable Language Tone:

✓ "You're currently safe" ✓ "You can afford to miss 6 more classes" ✓ "Recovery margin: comfortable"

✗ Avoid: "Great job!", "Keep it up!" (no cheerleading) Color Palette (Safe State):

css --bg: #FAFAFA; /* Soft off-white / _--text: #374151;_ / Warm dark grey / _--accent: #10B981;_ / Muted green (minimal use) */ --surface: #FFFFFF; --border: #E5E7EB; Animation Speed:

css .safe-state-transition { transition: all 600ms cubic-bezier(0.4, 0, 0.2, 1); }

### 🟡 STATE 2: TENSION

Condition:
- Buffer 2-4 classes
- OR approaching Point of No Return (within 14 days)
- Percentage 75-80%

UX Behavior:
- **Layout:** Slightly compressed (16px gaps), tighter grouping
- **Animation:** Faster (300ms), reduced easing (`ease-in-out`)
- **Interaction:** Some actions require confirmation ("Are you sure?")
- **Timeline:** Auto-focuses on near-term risks
- **Visual Feel:** Alert but not panicked, focused attention

Language Tone:

⚠️ "Be careful from here" ⚠️ "Skipping now reduces recovery options" ⚠️ "Buffer: 2 classes remaining"

✗ Avoid: "Warning!", "Danger!" (too alarming) Color Palette (Tension State):

css --bg: #FFFBEB; /* Warm cream / _--text: #1F2937;_ / Darker grey / _--accent: #F59E0B;_ / Amber/yellow / _--surface: #FEF3C7;_ / Light amber tint */ --border: #FCD34D; Animation Speed:

css .tension-state-transition { transition: all 300ms ease-in-out; } Interaction Changes:

jsx // Example: Skip button requires confirmation in Tension state {state === 'TENSION' && (Skip Next Class)}

// Confirmation modal appears "Skipping will reduce your buffer to 1 class. Continue?" [Cancel] [Skip Anyway]

### 🔴 STATE 3: CRITICAL

Condition:
- 1-2 absences can break 75%
- OR recovery no longer mathematically possible
- Buffer ≤ 1 or deficit exists

UX Behavior:
- **Layout:** Minimal UI, reduced elements, essential info only
- **Animation:** No decorative motion, only functional (150ms, linear)
- **Interaction:** Fewer actions available, simulations disabled (if recovery impossible)
- **Visual Feel:** Strong focus, high contrast, urgent but clear

Language Tone:

🚨 "You must attend all remaining classes" 🚨 "Recovery becomes impossible after 5 Feb 2026" 🚨 "One more skip = detention"

Direct, factual, non-emotional. No sugarcoating. No false hope. Color Palette (Critical State):

css --bg: #FEF2F2; /* Soft red tint / *--text: #111827;* / Near-black / *--accent: #DC2626;* / Muted red / *--surface: #FEE2E2;* / Light red surface */ --border: #EF4444; Animation Speed:

css .critical-state-transition { transition: all 150ms linear; /* Fast, no easing */ }

/* Pulse stops completely / *.health-ring.critical { animation: none;* / No pulse in critical state */ }

UI Reduction:

Safe State UI: Critical State UI:

- Health Ring - Health Ring (no pulse)
- Timeline - Timeline (locked)
- All subjects - Critical subjects only
- Skip simulator - Simulator DISABLED

- Recovery planner - "Recovery impossible" message
- 7-day planner - HIDDEN

Only essential, actionable information.

## 3. INFORMATION ARCHITECTURE (SINGLE VERTICAL FLOW)

### No Tabs. No Side Navigation. No Menu.

Vertical Flow Only:

```
┌──────────────────────────────────────────┐ │ 1. Global Attendance
Health Ring │ ← Always visible
├──────────────────────────────────────────┤ │ 2. Irreversibility Timeline │
← Core insight ├──────────────────────────────────────────┤ │ 3. Subject
Status Overview │ ← List of subjects │ (Strips, not cards) │
├──────────────────────────────────────────┤ │ 4. Focused Subject Detail
│ ← Inline expansion │ (When subject tapped) │
└──────────────────────────────────────────┘
```

Navigation is contextual:
- Tap subject → Expands inline
- Other subjects fade (focus isolation)
- Scroll to navigate
- No "Back" buttons (collapse by tapping again)

## 4. COMPONENT SPECIFICATIONS

### Component 1: Attendance Health Ring

Purpose: Immediate situational awareness

Visual Design:

```
╔══════════════════╗
║           ║
║     78%      ║ ← Large percentage (only shown on tap)
║           ║
║   TENSION     ║ ← State label
╚══════════════════╝
   Ring color adapts to state
```

Behavior:

javascript // Ring animation on load .health-ring { animation: ringGrow 800ms cubic-bezier(0.34, 1.56, 0.64, 1); }

@keyframes ringGrow { 0% { transform: scale(0.8); opacity: 0; } 100% { transform: scale(1); opacity: 1; } }

// Color transitions smoothly between states .health-ring.safe { stroke: #10B981; } .health-ring.tension { stroke: #F59E0B; } .health-ring.critical { stroke: #DC2626; }

// State-specific behavior .health-ring.safe { animation: subtlePulse 3s ease-in-out infinite; }

@keyframes subtlePulse { 0%, 100% { opacity: 1; } 50% { opacity: 0.8; } }

// Pulse STOPS in critical state .health-ring.critical { animation: none; /* No pulse = gravity of situation */ }

Interaction:

User taps ring → Tooltip appears:

"TENSION STATE ━━ You have 3 classes remaining before entering critical zone.

Your buffer is shrinking. Plan skips carefully from this point forward."

[Dismiss] Antigravy Component Spec:

jsx <HealthRing percentage={78.2} state="TENSION" // SAFE | TENSION | CRITICAL buffer={3} onTap={showStateExplanation} animationDuration={state === 'SAFE' ? 600 : state === 'TENSION' ? 300 : 150} /> CSS Variables (State-Adaptive):

css .health-ring { --ring-size: 160px; --ring-thickness: 12px; --ring-color: var(--state-color); transition: stroke var(--animation-speed) ease; }

/* Percentages HIDDEN by default */ .health-ring .percentage { opacity: 0; transition: opacity 200ms; }

.health-ring:hover .percentage, .health-ring.active .percentage { opacity: 1; }

### Component 2: Irreversibility Timeline

This is NOT a calendar. This is a constraint visualizer.

Shows:
- **Today** (current position)
- **Last Safe Skip** (last day you can skip without entering tension)
- **Recovery End Point** (Point of No Return date)
- **Semester End**

Visual Design:

Timeline (Horizontal scroll, mobile):

Safe Zone Tension Critical Locked
━━━━━━━━━━━━━━━━━━━━━━━━━━━━●━━━━━━━━━━━━━━━━━━●━━━━━━━━━━━━━━
━●━━━━━━━━━━━━━━━━━ Today ↑ Last PNR Semester Safe Skip End

Color gradient: Green → Yellow → Red → Grey (locked)

Width compression: Safe zone: Normal spacing Tension zone: 0.8× compressed Critical zone: 0.6× compressed (visual urgency) Locked zone: 0.4× compressed (impossibility) Behavior:

css /* Timeline auto-centers near next irreversible point */ .timeline-container { scroll-snap-type: x mandatory; overflow-x: auto; }

.timeline-point { scroll-snap-align: center; }

/* Critical zones visually compress space */ .timeline-zone.safe { width: 100%; } .timeline-zone.tension { width: 80%; filter: saturate(1.2); /* Slight color intensification */ } .timeline-zone.critical { width: 60%; filter: saturate(1.5); } .timeline-zone.locked { width: 40%; opacity: 0.5; pointer-events: none; }

/* Scroll resistance in CRITICAL state */ .timeline-container.critical { scroll-snap-type: none; /* Allow free scroll */ overscroll-behavior: contain; /* Add friction */ }

Labels:

"Last Safe Skip" (green dot) "Recovery Ends" (red dot, pulsing if within 7 days) "Locked Zone" (grey, with lock icon)

Interaction:

Tap timeline point → Shows detail:

"Last Safe Skip: 28 Jan 2026
── After this date,
every skip moves you into tension state.

Buffer remaining: 3 classes" Purpose:

"To make future constraints felt, not just known."

Antigravy Component Spec:

jsx <IrreversibilityTimeline today={new Date('2026-01-23')} lastSafeSkip={new Date('2026-01-28')} pnrDate={new Date('2026-02-05')} semesterEnd={new Date('2026-05-15')} currentState="TENSION" autoCenter="next-critical" // Auto-scroll to next critical point />

### Component 3: Subject Status Strips

NOT card-based. Thin horizontal strips.

Visual Design:

```
┌───────────────────────────────────────┐ │ 🟢 Machine Learning
68.2% │ ← Strip (collapsed)
├───────────────────────────────────────┤ │ 🟡 Data Analytics
77.5% │ ├───────────────────────────────────────┤ │ 🟢 Cloud
Computing 82.1% │ └───────────────────────────────────────┘
```

Tap to expand:

```
┌───────────────────────────────────────┐ │ 🔴 Machine Learning
68.2% │ ← Expanded strip
├───────────────────────────────────────┤ │
┌───────────────────────────────────┐ │ │ │ [Attendance Budget
Surface] │ │ ← Inline detail │ │ Attended ██████░░░░░░ Unsafe │ │ │ │ │ │ │ │
Buffer: 0 classes │ │ │ │ Deficit: 20 classes needed │ │ │ │ │ │ │ │ [What-If Simulation
Buttons] │ │ │ └───────────────────────────────────┘ │
├───────────────────────────────────────┤ │ Data Analytics 77.5%
│ ← Other subjects fade │ (faded, opacity: 0.4) │
└───────────────────────────────────────┘ Behavior:
```

css /* Default: Thin strips */ .subject-strip { height: 56px; padding: 12px 16px; border-bottom: 1px solid var(--border); transition: all var(--animation-speed); }

.subject-strip.expanded { height: auto; /* Expands inline */ background: var(--surface); box-shadow: 0 4px 12px rgba(0,0,0,0.08); }

/* Other subjects fade (focus isolation) */ .subject-strip:not(.expanded) { opacity: 0.4; filter: grayscale(0.3); }

/* No dense lists _/ .subject-strip_ + _.subject-strip { margin-top: 0; /_ No gaps, continuous flow */
}

Interaction:

Tap subject strip → Expands inline Tap again (or tap another) → Collapses Scroll → Other subjects visible but faded No tables. No card grids. No dense lists.

Antigravy Component Spec:

jsx <SubjectStatusStrips subjects={subjects} expandedSubject={expandedId} onExpand={(id) => setExpandedId(id)} focusIsolation={true} // Fades other subjects />

### Component 4: Attendance Budget Surface

Attendance is treated as a budget, not a percentage.

Visual Model:

Horizontal capacity surface:

Attended Remaining Unsafe Zone
████████████░░░░░░░░░░░▓▓▓▓▓▓▓▓

← Filled region ← Available ← Shaded, pushes inward (Dark) (Light) (Red tint)

As risk increases: Unsafe zone ▓▓▓ advances left → Creates visual pressure Behavior:

css .budget-surface { width: 100%; height: 40px; display: flex; border-radius: 8px; overflow: hidden; }

.budget-attended { background: linear-gradient(90deg, #374151, #4B5563); transition: width 300ms cubic-bezier(0.4, 0, 0.2, 1); }

.budget-remaining { background: #E5E7EB; transition: width 300ms; }

.budget-unsafe { background: repeating-linear-gradient(45deg, #FEE2E2, #FEE2E2 10px, #FCA5A5 10px, #FCA5A5 20px); transition: width 300ms; /* Pushes inward as risk increases */ }

/* Simulating absence → unsafe zone advances */ .budget-surface.simulating-skip .budget-unsafe { width: calc(var(--unsafe-width) + 5%); /* Grows */ animation: unsafeAdvance 300ms ease-out; }

@keyframes unsafeAdvance { 0% { transform: scaleX(1); } 50% { transform: scaleX(1.05); } /* Slight push */ 100% { transform: scaleX(1); } } Antigravy Component Spec:

jsx

### Component 5: What-If Simulation (CORE INTERACTION LOOP)

Interaction Model: Explicit actions, NOT sliders.

Buttons:

┌───┐ │ What-If Simulation │ ├───┤ │ │ │ [Miss Next Lecture] │ ← Explicit action │ [Attend All Upcoming Classes] │ │ [One Lecture Cancelled] │ │ │ │ [Reset Simulation] │ └───┘ Rules: Each action updates (in order):

Health Ring (color, percentage) Timeline (position shifts) Subject Budget Surface (unsafe zone advances/retreats) Critical: ~300ms delay between trigger and update

javascript // Delay represents decision weight const handleSkipSimulation = () => { setIsSimulating(true);

setTimeout(() => { updateHealthRing(); updateTimeline(); updateBudgetSurface(); setIsSimulating(false); }, 300); // Deliberate delay }; Why the delay?

"No instant jumps. The delay creates psychological weight—this is a consequential decision, not a casual toggle."

Visual Feedback During Delay:

css .simulation-button.active { animation: buttonPulse 300ms ease-out; }

@keyframes buttonPulse { 0% { transform: scale(1); } 50% { transform: scale(0.95); } 100% { transform: scale(1); } } Antigravy Component Spec:

jsx <WhatIfSimulation subject={currentSubject} onAction={(action) => simulateAction(action, 300)} // 300ms delay actions={[{ id: 'skip-next', label: 'Miss Next Lecture' }, { id: 'attend-all', label: 'Attend All Upcoming' }, { id: 'cancelled', label: 'One Lecture Cancelled' }]} delayMs={300} // Decision weight delay />

## 5. ATTENDANCE FORECAST & RECOVERY LOGIC

System Calculates:
- Maximum recoverable attendance
- Point where recovery becomes impossible (PNR)
- Required attendance per remaining class

Displayed As:
Direct statements, NOT formulas.

Example Outputs:

✓ Good: "Even if you attend all remaining classes, 75% cannot be reached.

Maximum achievable: 73.2%"

✗ Bad: "Required attendance rate: 0.89 Current trajectory: Below threshold Confidence interval: 95%"

Recovery Plan Format:

┌──────────────────────────────────────┐ │ Recovery Plan: Machine Learning │ ├──────────────────────────────────────┤ │ │ │ Current: 68.2% │ │ Target: 75.0% │ │ │ │ Required: Attend 22 of next 28 │ │ │ │ Week 3-6: Attend ALL (4/week) │ ← Direct instructions │ Week 7-10: Can miss max 1/week │ │ Week 11+: Can miss max 1/week │ │ │ │ Result: 75.8% achievable ✓ │ │ │ │ Difficulty: HARD │ │ │ └──────────────────────────────────────┘

If Recovery Impossible:

┌──────────────────────────────────────┐ │ Machine Learning │ ├──────────────────────────────────────┤ │ │ │ Recovery Status: IMPOSSIBLE │ │ │ │ Even if you attend all 39 remaining │ │ classes, you will reach only 73.2%. │ │ │ │ 75% is no longer mathematically │ │ achievable for this subject. │ │ │ │ Next steps: │ │ • Talk to faculty about condonation │ │ • Check medical certificate policy │ │ │ │ [Contact Faculty Advisor] │ │ │ └──────────────────────────────────────┘ No sugarcoating. No false hope.

6. ANIMATION SYSTEM (STRICT RULES) Allowed Animations: ✅ State transitions (Safe → Tension → Critical) ✅ Risk changes (buffer decreasing) ✅ Simulation consequences (budget surface shifting) ✅ Timeline scrolling to critical points

Forbidden Animations: ❌ Hover-only animations (decorative) ❌ Bounce/elastic easing (playful tone) ❌ Cosmetic motion (page transitions, confetti) ❌ Loading spinners (use skeleton states instead)

Rule: "Animation must explain risk, relief, or irreversibility."

Example - Allowed:

css /* State transition (explains change) */ .health-ring { transition: stroke 600ms ease-in-out; }

/* Buffer decreasing (shows risk) */ .budget-unsafe { animation: unsafeAdvance 300ms ease-out; } Example - Forbidden:

css /* Hover bounce (decorative) / _.button:hover { animation: bounce 500ms;_ / ❌ NO */ }

/* Page transition (cosmetic) / _.page-enter { animation: slideIn 800ms cubic-bezier(0.68, -0.55, 0.265, 1.55);_ / ❌ NO */ } Animation Speed by State:

javascript const animationDurations = { SAFE: 600, // Slow, relaxed TENSION: 300, // Medium, alert CRITICAL: 150 // Fast, minimal }; 7. COLOR SYSTEM (ANTI-VIBE-CODING) Base Palette (State-Agnostic) css /* Foundation colors / _--warm-dark-grey: #374151; --soft-off-white: #FAFAFA; --near-black: #111827; --pure-white: #FFFFFF; Accent (ONE COLOR ONLY) css_ / State determines accent color / _--accent-safe: #10B981;_ / Muted green / _--accent-tension: #F59E0B;_ / Amber / _--accent-critical: #DC2626;_ / Muted red */

/* Current accent (set by state) */ --accent: var(--accent-tension); Color Usage Rules: Color is used ONLY to signal risk changes No persistent blue themes (not a productivity app) Backgrounds adapt to state (cream in tension, soft red in critical) Text always high contrast (4.5:1 minimum) State-Specific Palettes:

css /* Safe State */ .state-safe { --bg: #FAFAFA; --surface: #FFFFFF; --text: #374151; --accent: #10B981; --border: #E5E7EB; }

/* Tension State / _.state-tension { --bg: #FFFBEB;_ / Warm cream / _--surface: #FEF3C7;_ / Light amber / _--text: #1F2937; --accent: #F59E0B;_ / Amber */ --border: #FCD34D; }

/* Critical State / _.state-critical { --bg: #FEF2F2;_ / Soft red tint / _--surface: #FEE2E2; --text: #111827;_ / Near-black / _--accent: #DC2626;_ / Muted red */ --border: #EF4444; }

### No Rainbow Dashboards

✗ Bad: Blue for ML, Green for DAV, Purple for CSS...

✓ Good: All subjects use state color (green/yellow/red) 8. TYPOGRAPHY & LANGUAGE Font System css /* Single serious sans-serif */ font-family: 'Inter', -apple-system, system-ui, sans-serif;

/* Hierarchy _/ --text-display: 32px; /_ Health ring percentage _/ --text-heading: 20px; /_ Section titles _/ --text-body: 16px; /_ Default text _/ --text-small: 14px; /_ Labels, metadata */

/* Weight */ --weight-normal: 400; --weight-medium: 500; --weight-semibold: 600;

/* Numbers slightly heavier _/ .percentage, .buffer-count, .deficit-count { font-weight: 600; font-variant-numeric: tabular-nums; /_ Aligned numbers */ }

### Language Rules

Direct, non-friendly language:

✓ **Use:**
- "You can afford to miss…"
- "After this point, recovery is impossible"
- "Buffer: 3 classes"
- "Attend all remaining classes"
- "One more skip = detention"

✗ **Avoid:**
- "Insights" (too vague)
- "Analytics" (not a dashboard)
- "Smart suggestions" (not AI)
- "Great job!" (no cheerleading)
- "Oops!" (too casual)
- "Don't worry" (patronizing)

Tone Comparison:

Generic App: 75Guard: "You're doing great!" "You're currently safe" "Uh oh, low attendance" "Buffer: 1 class remaining" "Try to attend more" "You must attend all remaining classes"

## 9. EDGE & FAILURE STATES

### If Recovery Impossible:

UI Changes:
- Simulations DISABLED (buttons greyed out)
- Message shown clearly (no modals)
- UI enters locked CRITICAL mode
- Timeline shows "Locked Zone"
- Recovery planner replaced with next steps

Example:

┌─────────────────────────────────────┐ │ 🔴 Machine Learning 62.5% │ │ ├─────────────────────────────────────┤ │ │ │ RECOVERY NO LONGER POSSIBLE │ │ │ │ Remaining classes: 18 │ │ Maximum achievable: 71.7% │ │ │ │ 75% cannot be reached even with │ │ perfect attendance. │ │ │ │ What you can do: │ │ • Request faculty condonation │ │ • Submit medical certificates │ │ • Review attendance policy │ │ │ │ [Contact Faculty Advisor] │ │ [View Attendance Policy] │ │ │ └─────────────────────────────────────┘ No sugarcoating. No false hope.

Simulation buttons when recovery impossible:

html Miss Next Lecture

Simulations disabled: Recovery no longer possible

10. WHAT THIS UI MUST NOT BECOME Anti-Patterns to Avoid: ❌ A generic SaaS dashboard

No "Welcome back, [Name]!" No greeting cards No "You have X notifications" ❌ A student ERP clone

No tables with 10+ columns No admin-style navigation No "Export to Excel" everywhere ❌ A colorful productivity app

No confetti animations No achievement badges No gamification (streaks, points) ❌ A percentage calculator

No input fields for "What percentage do you want?" No pie charts of attendance No comparison graphs What It Should Be: ✅ A decision surface, not an information dump ✅ Focused on one question: What happens if I skip? ✅ Constraint-aware, not feature-rich ✅ Serious, not playful

11. RESPONSIVE BEHAVIOR (MOBILE-FIRST) Breakpoints css /* Mobile-first (default) */ .container { padding: 16px; max-width: 100%; }

/* Tablet (768px+) */ @media (min-width: 768px) { .container { padding: 24px; max-width: 720px; margin: 0 auto; }

/* Timeline becomes horizontal */ .timeline { flex-direction: row; } }

/* Desktop (1024px+) */ @media (min-width: 1024px) { .container { max-width: 960px; }

/* Subjects can show 2 columns in safe state only */ .state-safe .subject-strips { display: grid; grid-template-columns: repeat(2, 1fr); gap: 16px; }

/* Critical state: Always single column */ .state-critical .subject-strips { display: block; } }

### Mobile Gestures

Swipe down → Collapse expanded subject Swipe up → Expand subject (alternative to tap) Pinch on timeline → Zoom to critical zone 12. ACCESSIBILITY (A11y) REQUIREMENTS Color Independence html

🔴 CRITICAL Continue

7:41 PM Machine Learning requires immediate attention

```` Keyboard Navigation Tab: Navigate between subjects Enter: Expand/collapse subject Space: Activate simulation button Esc: Close expanded view Arrow keys: Navigate timeline Screen Reader Support html
>
78.2%

Your overall attendance is 78.2%, currently in tension state. You have 3 classes remaining before entering critical zone.

Focus Indicators css /* High contrast focus states */ button:focus, .subject-strip:focus { outline: 3px solid var(--accent); outline-offset: 2px; }

/* Never remove outlines */ *:focus { outline-style: solid; /* Enforced */ } 13. PERFORMANCE REQUIREMENTS Load Time Initial render: <500ms State calculation: <100ms Simulation update: 300ms (deliberate delay) Timeline scroll: 60fps Optimization javascript // Memoize expensive calculations const calculatedState = useMemo(() => { return calculateAttendanceState(subjects, semester); }, [subjects, semester]);

// Debounce timeline scroll const handleTimelineScroll = debounce(() => { updateVisibleRange(); }, 100); Bundle Size Target Total JS: <150KB gzipped CSS: <30KB gzipped Fonts: Self-hosted, <50KB 14. ANIMATION SPECIFICATIONS (COMPLETE) State Transition Animations css /* Safe → Tension */ @keyframes safeToTension { 0% { background: #FAFAFA; --accent: #10B981; } 100% { background: #FFFBEB; --accent: #F59E0B; } }

/* Tension → Critical */ @keyframes tensionToCritical { 0% { background: #FFFBEB; --accent: #F59E0B; } 50% { background: #FEF2F2; } 100% { background: #FEF2F2; --accent: #DC2626; } }

/* Apply based on state change */ .state-container { animation: var(--state-transition) 600ms ease-in-out; } Health Ring Pulse (Safe State Only) css @keyframes subtlePulse { 0%, 100% { opacity: 1; transform: scale(1); } 50% { opacity: 0.85; transform: scale(1.02); } }

.health-ring.safe { animation: subtlePulse 3000ms ease-in-out infinite; }

.health-ring.tension, .health-ring.critical { animation: none; /* Pulse stops in risk states */ } Budget Surface Simulation css @keyframes unsafeAdvance { 0% { width:
````

var(--unsafe-width); } 50% { width: calc(var(--unsafe-width) + 5%); filter: brightness(1.1); } 100% { width: calc(var(--unsafe-width) + 5%); } }

.budget-unsafe.simulating { animation: unsafeAdvance 300ms ease-out forwards; } Timeline Auto-Scroll javascript // Smooth scroll to next critical point timelineContainer.scrollTo({ left: criticalPointX, behavior: 'smooth' });

// Scroll speed adapts to state const scrollDuration = { SAFE: 800, TENSION: 500, CRITICAL: 300 }; 15. COMPONENT INTERACTION MAP User Action → UI Updates → Feedback

---

──────────────── Open app → Calculate state → Render appropriate UI → Health ring animates in
→ Timeline centers on today

Tap Health Ring → Show state explanation → Tooltip with delay (200ms) → Subtle scale animation

Tap Subject Strip → Expand inline → 300ms ease-out → Other subjects fade (0.4) → Focus isolation → Budget surface renders

Tap "Miss Next Lecture" → Button pulse (150ms) → Visual feedback → Wait 300ms (decision weight) → Pause → Update health ring → Color transition → Update timeline → Position shift → Update budget surface → Unsafe zone advances → Show new state → Numbers update

Scroll Timeline → Update visible range → 60fps smooth → Highlight nearest point → Auto-snap (optional) → Compress critical zones → Visual urgency

State changes → Background color transition → 600ms ease (Safe → Tension) → Accent color updates → All elements adapt → Animation speed changes → Feels different → Spacing compresses → Layout shifts 16. DEVELOPMENT HANDOFF CHECKLIST Design Assets Required: Figma file with all 3 states (Safe, Tension, Critical) Component library (Health Ring, Timeline, Strips, Budget Surface) Icon set (status dots, timeline markers, lock icon) Color variables (CSS custom properties) Typography scale and weights Animation specifications (durations, easings) Mobile, tablet, desktop mockups Empty states and error states Accessibility annotations Developer Documentation: State calculation logic (pseudocode) Component interaction flowchart Animation timing reference Responsive breakpoint guide Keyboard navigation map ARIA label requirements Testing Requirements: All 3 states render correctly State transitions smooth (no jumps) Simulations update all components Timeline scrolls to critical points Mobile gestures work (swipe, tap, pinch) Keyboard navigation complete Screen reader announces state changes High contrast mode compatible Loads in <500ms on 3G Works offline (PWA) 17. FINAL DESIGN PRINCIPLES SUMMARY State-Driven, Not Screen-Driven UI adapts completely to Safe/Tension/Critical state No neutral home screen Constraint Visualization Timeline makes future limits visible Budget surface shows capacity shrinking Animations create psychological weight Focus Isolation Expand subject → others fade Critical state → minimal UI One thing at a time Deliberate Friction 300ms delay on simulations Confirmations in tension state Scroll resistance near PNR Direct Language No euphemisms No cheerleading Facts, consequences, clarity Animation as Communication Every motion explains risk Pulse stops in critical (gravity) Speed adapts to urgency Anti-Dashboard Not a browsing interface Not an information dump A decision instrument