

Supplementary Material S4. Runtime Behavior Pseudocode

The study uses CSS code exported from Figma Design via the “Copy as CSS” function. The exports describe the visual design but not the responsive-layout logic or JavaScript event handlers. Indicators V2 (Text Scaling), O2 (Pointer Cancellation) and A3 (Error Prevention) involve runtime behavior, which CSS alone cannot supply; their scoring follows the pseudocode below. The pseudocode is the contract the prototype must honor at deployment, and is used as evaluation input together with the CSS and the screenshots.

Pseudocode conventions. The blocks below follow standard CSS and ECMAScript syntax. Identifiers used but not declared inside the blocks (e.g., `executeAction`, `abortAction`, `showConfirmDialog`, `createToast`, `rollbackLastAction`, `computeImpactPreview`, `adjustChairPosition`, `startCountdown`, `placeCall`, `showCancelOption`) denote application-level abstractions supplied by the host application; their concrete implementations are not in scope for this evaluation contract.

S4.1. Responsive Adaptation Pseudocode (V2 Text Scaling)

```
html { font-size: 16px; }
```

```
.scenario-container {  
  width: 100%;  
  max-width: 23.4375rem; /* 375px / 16 */  
  margin: 0 auto;  
  position: relative;  
}
```

```
.card {  
  display: flex;  
  flex-direction: column;  
  min-height: auto;  
  padding: 1rem;  
}
```

```
.heading-1 { font-size: 1.75rem; } /* 28px */  
.heading-2 { font-size: 1.5rem; } /* 24px */  
.body { font-size: 1.25rem; } /* 20px */  
.caption { font-size: 0.875rem; } /* 14px */
```

```
/* Older-adult mode: class toggled by the host application based on user preference. */  
html.older-adult-mode { font-size: 20px; }
```

Under 200% user-level zoom (equivalent to font-size: 32px), all text scales proportionally; containers expand via flex with min-height: auto, so neither truncation nor overlap occurs. The older-adult-mode class is toggled by the host application according to the user’s preferred reading size and increases the root font-size from 16 px to 20 px while preserving the rem-based

proportional sizing of all descendants.

S4.2. Pointer Cancellation Event Binding (O2)

```
interactiveElement.addEventListener('touchstart', (e) => {  
  delete e.target.dataset.touchCancelled;  
});
```

```
interactiveElement.addEventListener('touchend', (e) => {  
  if (!e.target.dataset.touchCancelled) {  
    executeAction(e.target);  
    showUndoToast(5000);  
  }  
});
```

```
interactiveElement.addEventListener('touchcancel', (e) => {  
  e.target.dataset.touchCancelled = 'true';  
  abortAction(e.target);  
});
```

```
interactiveElement.addEventListener('touchmove', (e) => {  
  const touch = e.touches[0];  
  const rect = e.target.getBoundingClientRect();  
  if (touch.clientX < rect.left || touch.clientX > rect.right ||  
      touch.clientY < rect.top || touch.clientY > rect.bottom) {  
    e.target.dataset.touchCancelled = 'true';  
  }  
});
```

```
function showUndoToast(duration) {  
  const toast = createToast('Action executed; tap to undo');  
  toast.onclick = () => rollbackLastAction();  
  setTimeout(() => toast.dismiss(), duration);  
}
```

The pseudocode jointly implements the three mechanisms required by WCAG SC 2.5.2: up-event activation, gesture abort, and undo. The initial touchstart handler clears the touchCancelled flag at the start of every touch, so a cancellation in one interaction does not bleed into the next.

S4.3. High-Risk Operation Protection (A3)

```
function triggerHighRiskAction(action) {  
  const affectedDevices = computeImpactPreview(action);  
  const confirmText =  
    `About to execute "${action.name}", ` +  
    `affecting ${affectedDevices.length} device(s): ` +  
    `${affectedDevices.join(', ')} `;
```

```

showConfirmDialog({
  text: confirmText,
  onConfirm: () => {
    executeAction(action);
    showUndoToast(action.undoWindow || 9000);
  },
  onCancel: () => { /* no-op */ }
});
}

```

```

// Assistive chair stand/sit: release-to-stop
assistiveChairButton.addEventListener('touchstart', (e) => {
  const interval = setInterval(() => adjustChairPosition(e.target), 50);
  const stop = () => clearInterval(interval);
  e.target.addEventListener('touchend', stop, { once: true });
  e.target.addEventListener('touchcancel', stop, { once: true });
});

```

```

// Emergency call: 9-second countdown undo window
function initiateEmergencyCall(contact) {
  startCountdown(9, () => placeCall(contact));
  showCancelOption(contact);
}

```

High-risk operations jointly satisfy the three mechanisms required by WCAG SC 3.3.4: reversibility, checkability, and confirmation. Daily control pages without high-risk operations (the lighting, air-conditioning, and bedroom control pages) are not in scope for Section S4.3 and are handled by Precedence Rule 3 of Section S3.5 ($S_i = 100$, “no applicable target”).