Accessibility Conformance Report (VPAT 2.4), Decision Art

Product: Art & Mind on Decision Art (decisionart.app), all student reading surfaces **Version evaluated:** Production as of 2026-10-17 **Report date:** 2026-10-17 **Prepared for:** Institutional accessibility reviewers **Contact:** Dan Hahn, dan@aportlandcareer.com **VPAT version:** 2.4 (WCAG edition) **Standards evaluated:** WCAG 2.1 Level A and Level AA

A Section 508 edition and an EN 301 549 edition of this report can be supplied on request; the underlying evaluation is the same.

This edition updates our report of 2026-09-17 with a re-test on 2026-10-17. It is a good-faith self-assessment. It declares its remaining gaps rather than claiming a clean sweep.

Scope of this evaluation

This report evaluates **every surface on which a student reads the book**, under a single standard:

- the **course reader** reached when a student launches Decision Art from Canvas (reading, annotation and in-text discussion, activities, gradebook), and
- the **standalone reader** used by someone who purchased the book directly.

The previous report evaluated only the course reader and declared the standalone reader weaker and out of scope. On re-examination that exclusion was wrong rather than necessary: both readers render the same viewer component, and the accessible reading mode (real reflowable text, adjustable size and spacing, three color themes, human-authored alt text, read-aloud) was already present and ungated on both. The exclusion described our previous review coverage, not the product. Both surfaces are therefore evaluated together here, and there is no scope exception to rule on.

Instructor-only management screens are included where a criterion is affected, and noted as such.

Evaluation methods

1. **Source-code review** of the React/Next.js components: ARIA usage, keyboard handling, focus management, and the text-to-speech and accessible-text pipeline.
2. **Measured color contrast.** An automated script computes contrast for the design tokens and every foreground/background pair in use, and **fails the build** when a pair falls below WCAG AA. It runs in `prebuild` and in the project's `check` task, so a failing pair stops a release rather than producing a warning somebody has to notice. It is protected by its own test suite. When first run it caught nine failing pairs, including dark-mode primary buttons at 2.66:1, all since corrected.
3. **Automated accessibility sweep (axe), across signed-out and signed-in pages.** The sweep previously covered public pages only, which meant it described the marketing site rather than the

product. It now also runs under real student and instructor sessions and covers the standalone reader, the course portal, the course pages, the student scores page, the instructor gradebook, the people manager, and the accessible-text reading mode, which are the surfaces this review concerns.

Two things about how it is built are worth stating, because they are what makes its result meaningful. It refuses to report on a page that rendered signed out, or that was still showing a loading placeholder, since both of those scan clean and would turn the whole sweep into a green light that measured nothing. And it carries a student session for student pages and an instructor session for instructor pages, because a student landing on a gradebook gets a refusal page that axe is also happy to call clean.

Re-test, 2026-10-17. The signed-out and signed-in sweeps, with the resize, reflow, text-spacing and hover-or-focus checks, ran against the current build: 89 checks passed, none failed and none were skipped. The re-test added a guard that refuses to score a page that turned the reviewer away ("Not authorized"), since such a page also scans clean. It found and fixed two Level A issues on instructor screens (unlabelled date fields on the deadlines page, and list items outside a list on the exercises page).

On its first fully authenticated run, in September, it found a Level AA failure that none of our other checks could see: on the student scores page, muted text at 60% opacity measured 2.45:1 against white, where AA requires 4.5:1. The design token was correct; a utility class halved it when painted, which is exactly the gap between a static palette check and a measurement of the rendered page. It is fixed, and the sweep now passes on every surface listed above. An earlier axe pass over the reader had found ten further issues, two of them systemic (every slider in the application was unnamed, and closing any dialog dropped focus to the document body); those are fixed as well.

4. **Keyboard checks in the automated browser tests:** dialogs move focus in, trap it, close on Escape and return focus; tooltips appear on keyboard focus and close on Escape without moving it. A manual screen-reader spot check has not yet been recorded; it is listed under remaining issues.

Changes since the previous report

Since the September edition, three more items are closed: every remaining hand-rolled overlay panel now uses the accessible dialog primitive; accessibility linting (`eslint-plugin-jsx-a11y` , the full recommended set) runs as errors on every change; and autocomplete tokens are set on every email and name field. The September edition's four closed items follow.

1. **Modal dialogs in the reading and grading flow** now use an accessible dialog primitive with `role="dialog"` , `aria-modal` , focus trapping, focus moved in on open and restored on close, and Escape to close. The non-keyboard trigger is now a native button. Eight secondary panels, mostly instructor-side viewers, were still hand-rolled overlays in September; all have since been converted. (1.3.1, 2.1.1, 2.4.3, 4.1.2)
2. **Accessible-text headings** now emit real semantic heading levels rather than geometry-inferred paragraphs. (1.3.1, 2.4.6)
3. **Color contrast** is now measured and enforced in the build rather than asserted. (1.4.3, 1.4.11)

4. **The application-wide 125% zoom has been removed.** A single CSS rule previously scaled the whole application to 125%, which multiplied with the reader's own browser zoom, so a user at 110% was effectively reading at 137% and reaching the 200% requirement much sooner than the browser reported. It is gone, along with the tooltip positioning compensation it forced. Criterion 1.4.4 has been rechecked against the current build rather than carried forward from the previous report.

Notable strengths

- **Built-in read-aloud** with real, labeled controls, a live "reading page N" status region, and click-a-sentence-to-start. It runs entirely on the student's device.
- **A dedicated accessible-text mode** that reflows the source PDF into navigable, real HTML text with adjustable size, line and letter spacing, column width, and three color themes (light, sepia, dark). Now available on both the course reader and the standalone reader.
- **Human-authored alt text** for the book's illustrations, written by the author rather than generated, exposed to screen readers in accessible-text mode.
- **Skip-to-content** link, semantic landmarks, native buttons and links with visible focus rings, decorative icons marked `aria-hidden`.
- **Accessible component primitives** (Radix) for dialog, tooltip, tabs, slider, and checkbox.

Equally effective alternate access

One criterion below (1.4.10 Reflow) is partially supported because the fixed PDF page view reproduces the printed page and does not reflow. The accessible-text mode is the equally effective alternative: it presents the same content as real reflowable text with user control over size, spacing, width, and color, plus read-aloud and authored alt text. It is available on every reading surface, is reachable by keyboard from the reader toolbar, and requires no request, no accommodation paperwork, and no separate file. We are glad to formalize this as an Equally Effective Alternate Access Plan with your accessibility office if that is the preferred instrument.

Equally effective means the graded work too, and we say so because we got it wrong once. Until 2026-09-18 the annotation sidebar rendered only in the page-image view, so a student reading in accessible-text mode could not see classmates' notes, could not reply, and could not write a note. Comments count toward the grade in courses that use them, which meant the alternate access path silently excluded a student from graded participation: the same content, but not the same functionality, which is not equal access. It is fixed. Reading the discussion, replying, voting, and composing a new note on a selected passage all work in accessible-text mode, and a reply written there is scored by exactly the same path as one written in the page view. We record it here rather than quietly repairing it because it is the kind of gap a conformance report exists to surface, and because it defines the standard we hold ourselves to: a criterion is not met by an alternative that presents the reading and withholds the coursework.

Remaining issues and remediation roadmap

1. **Third-party video captions (1.2.2).** Course videos are third-party embeds. Captions depend on the hosting platform and are not authored by the application. Remediation: require captioned sources for courses and show a caption status note to the instructor.
2. **Fixed PDF page view does not reflow (1.4.10, 1.4.5).** Covered by the alternate access above.
3. **Programmatic error association (3.3.3).** Errors are announced via live regions; field-level `aria-describedby` and `aria-invalid` association is not yet complete.
4. **Color-only signaling (1.4.1).** Interactive state uses `aria-pressed` and labels in addition to color, and the automated sweep found no violations, but automated tooling cannot fully audit color-only signaling. A manual pass is pending.
5. **Screen-reader spot check.** No dated manual check with VoiceOver or NVDA is recorded yet. A written script for one exists; until it is run, this report claims only what the automated checks show.

Table 1: WCAG 2.1 Level A

Criterion	Conformance	Remarks
1.1.1 Non-text Content	Partially Supports	Decorative icons are `aria-hidden`; icon-only buttons have accessible names; book illustrations have human-authored alt text in accessible-text mode. The native PDF page image relies on accessible-text mode for equivalents.
1.2.1 Audio-only and Video-only (Prerecorded)	Not Applicable	No first-party prerecorded audio-only or video-only content.
1.2.2 Captions (Prerecorded)	Partially Supports	Course videos are third-party embeds with an iframe title; captions depend on the hosting platform. See remaining issue 1.
1.2.3 Audio Description or Media Alternative	Not Applicable	No first-party video requiring audio description.
1.3.1 Info and Relationships	Supports	Landmarks, labeled form fields, ARIA groups, and live regions throughout, and accessible-text mode emits real heading levels. Every dialog, including the formerly hand-rolled panels, uses the accessible dialog primitive. Re-tested 2026-10-17.
1.3.2 Meaningful Sequence	Supports	Reading and DOM order follow the visual and logical order; accessible-text mode presents a linear reading sequence.
1.3.3 Sensory Characteristics	Supports	Instructions do not rely solely on shape, size, or position.

Criterion	Conformance	Remarks
1.4.1 Use of Color	Partially Supports	State is conveyed by `aria-pressed` and accessible names in addition to color; annotation highlights carry descriptive labels. A manual color-only audit is pending. See remaining issue 4.
1.4.2 Audio Control	Supports	Read-aloud provides pause, resume, and stop; no audio auto-plays.
2.1.1 Keyboard	Supports	Controls are native buttons and links, keyboard operable, with a skip link. The previously noted `div` trigger and Enter-only overlay are now native and handle both Enter and Space.
2.1.2 No Keyboard Trap	Supports	No keyboard traps identified. Dialog focus trapping releases correctly on close.
2.1.4 Character Key Shortcuts	Not Applicable	No single-character key shortcuts.
2.2.1 Timing Adjustable	Supports	No user-facing time limits on reading or activities.
2.2.2 Pause, Stop, Hide	Supports	Read-aloud is user-controlled; no auto-updating or auto-scrolling content.
2.3.1 Three Flashes or Below	Supports	No flashing content.
2.4.1 Bypass Blocks	Supports	Skip-to-content link targets the main region.
2.4.2 Page Titled	Supports	Descriptive page titles via framework metadata.
2.4.3 Focus Order	Supports	Focus order is logical in page content, and dialogs move focus in on open and restore it to the triggering control on close, as the automated dialog tests check. Re-tested 2026-10-17.
2.4.4 Link Purpose (In Context)	Supports	Link and button text is descriptive in context.
2.5.1 Pointer Gestures	Supports	No multipoint or path-based gestures required. Text selection for annotation has a keyboard equivalent.
2.5.2 Pointer Cancellation	Supports	Actions fire on the up-event via native controls.
2.5.3 Label in Name	Supports	Accessible names include the visible label text.
2.5.4 Motion Actuation	Supports	No motion-actuated functionality.
3.1.1 Language of Page	Supports	UI is English; the root layout sets `html lang="en"`.
3.2.1 On Focus	Supports	Focus does not trigger unexpected context changes.

Criterion	Conformance	Remarks
3.2.2 On Input	Supports	Input does not trigger unexpected context changes.
3.3.1 Error Identification	Supports	Form errors are announced with `role="alert"`.
3.3.2 Labels or Instructions	Supports	Inputs use associated labels; instructions are provided where needed.
4.1.1 Parsing	Supports	React renders well-formed markup.
4.1.2 Name, Role, Value	Supports	Components expose correct name, role, and value; all sliders carry accessible names, and every dialog exposes its role and name. Re-tested 2026-10-17.

Table 2: WCAG 2.1 Level AA

Criterion	Conformance	Remarks
1.2.4 Captions (Live)	Not Applicable	No live media.
1.2.5 Audio Description (Prerecorded)	Not Applicable	No first-party prerecorded video.
1.3.4 Orientation	Supports	Responsive layout works in portrait and landscape.
1.3.5 Identify Input Purpose	Supports	Autocomplete tokens are set on every email and name field.
1.4.3 Contrast (Minimum)	Supports	Measured two ways: a script that fails the build when any design token or foreground/background pair falls below AA, including the three accessible-text themes in light and dark mode, and an axe sweep of the rendered authenticated pages, which additionally catches colour composed at paint time (opacity and layering) that a token check cannot see.
1.4.4 Resize Text	Supports	Rechecked against the current build after the application-wide 125% zoom was removed. Content reflows and remains functional at 200% browser zoom without loss of content or functionality; accessible-text mode additionally offers its own text-size control.
1.4.5 Images of Text	Partially Supports	The source book is a PDF, which is images of text. Accessible-text mode provides a real-text equivalent on every reading surface.
1.4.10 Reflow	Partially Supports	Accessible-text mode reflows to a single column at 320 CSS pixels without two-dimensional scrolling. The fixed PDF page view reproduces the printed page and does not reflow. See "Equally effective alternate access" above.
1.4.11 Non-text Contrast	Supports	UI component boundaries, focus indicators, and graphical objects are covered by the same measured contrast check and the automated sweep.

Criterion	Conformance	Remarks
1.4.12 Text Spacing	Supports	Accessible-text mode exposes line and letter spacing controls; no clipping or overlap at increased spacing.
1.4.13 Content on Hover or Focus	Supports	Tooltips are dismissable, hoverable, and persistent, confirmed for the information tooltips in the authenticated sweep (2026-10-17). Chart tooltips are not separately tested.
2.4.5 Multiple Ways	Supports	Content is reachable via navigation, the table of contents, and the course home index.
2.4.6 Headings and Labels	Supports	Form labels and section headings are descriptive; accessible-text mode now emits real heading levels.
2.4.7 Focus Visible	Supports	Visible focus indicators on interactive elements, including within dialogs.
3.1.2 Language of Parts	Not Applicable	Single-language (English) content.
3.2.3 Consistent Navigation	Supports	Navigation is consistent across pages and across both readers.
3.2.4 Consistent Identification	Supports	Components are identified consistently.
3.3.3 Error Suggestion	Partially Supports	Errors are announced and suggest corrections; programmatic field-to-error association is pending. See remaining issue 3.
3.3.4 Error Prevention (Legal, Financial, Data)	Partially Supports	Payments use Stripe Checkout with its own confirmation step; in-app destructive actions confirm before proceeding, and a review of the remainder is pending.
4.1.3 Status Messages	Supports	Live regions announce reading status, save state, and errors without moving focus.

Statement

Decision Art is committed to WCAG 2.1 Level AA conformance across every surface on which a student reads the book. Since the previous report we have given the reading and grading dialogs proper dialog semantics and focus management, emitted real heading structure in the accessible-text reading mode, put color contrast under a measured check that fails the build, removed the application-wide zoom that compromised text resizing, and put automated accessibility testing behind the login where the substance of the product lives. We also corrected the previous report's scope: the accessible reading mode was already present on both readers, so the exclusion it declared described our review coverage rather than the product.

We welcome specific findings from any accessibility office and will prioritize them. Where a gap remains, we would rather state it here and agree a plan than have it found later.

Prepared by Decision Art. Contact: Dan Hahn, dan@aportlandcareer.com, https://decisionart.app