

drop-accessibility-spec

Drop Accessibility Audit Specification (WCAG 2.1 AA)

Version: 1.0 **Date:** 2026-02-17 **Project:** Drop — Fintech Payment App **Compliance Standard:** WCAG 2.1 Level AA + Norwegian Universal Design Law **Author:** Architect Agent **Status:** Draft

Executive Summary

This specification defines the accessibility audit requirements for Drop, a fintech payment application serving all residents of Norway and Scandinavia. The audit ensures compliance with:

- WCAG 2.1 Level AA** — International accessibility standard
- Norwegian Law:** Likestillings- og diskrimineringsloven § 18 (Equality and Anti-Discrimination Act)
- IKT-forskriften** — Regulation for Universal Design of ICT Solutions
- Digitaliseringsdirektoratet (Digdir) Guidelines** — Norwegian government digital accessibility requirements

Legal Context: Norway requires universal design of digital services for both public and private sectors. Non-compliance can result in fines from Digitaliseringsdirektoratet. WCAG 2.1 AA is the legal minimum standard in Norway.

Tech Stack: Drop uses Next.js 16 + React 19 + Radix UI (which has built-in accessibility features) + Tailwind v4. The application is mobile-first, with most users on phones.

1. Norwegian Legal Requirements

1.1 Applicable Laws

Law/Regulation	Section	Requirement
----------------	---------	-------------

Likestillings- og diskrimineringsloven	§ 18	Prohibition of discrimination based on disability; duty to provide universal design
Forskrift om universell utforming av IKT	All sections	Universal design of ICT solutions for public and private sectors (education, banking, commerce)
Finanstilsynet Guidelines	N/A	Financial services must be accessible to all customers
GDPR (DSGVO)	Article 25	Privacy by design — accessible consent forms and data access requests

1.2 Regulatory Bodies

- **Digitaliseringsdirektoratet (Digdir):** Supervises universal design of ICT
- **Likestillings- og diskrimineringsombudet (LDO):** Investigates discrimination complaints
- **Finanstilsynet:** Financial services regulator (monitors customer protection)

1.3 Compliance Deadlines

- **2025 Goal:** Norway universally designed by 2025 (national action plan)
- **Drop Target:** Full WCAG 2.1 AA compliance before public launch (Q2 2026)
- **Annual Monitoring:** Digdir annually monitors compliance — results reported to ESA/EU

1.4 Enforcement

- **Fines:** Digitaliseringsdirektoratet can issue fines for non-compliance
- **Discrimination Claims:** Users can file complaints with LDO
- **Reputational Risk:** Public disclosure of accessibility failures impacts trust in fintech sector

2. WCAG 2.1 AA Success Criteria Checklist

WCAG 2.1 AA includes **all 50 success criteria from WCAG 2.0 Level A/AA** plus **17 new criteria from WCAG 2.1**. Below is the complete checklist mapped to Drop features.

2.1 Principle 1: Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1: Text Alternatives

ID	Criterion	Level	Drop Application
1.1.1	Non-text Content	A	All images, icons, QR codes, logos have alt text or ARIA labels

Drop Implementation:

- Drop logo: `<img alt="Drop logo - green rounded square with dollar icon" />`
- Country flags in recipient list: `<span role="img" aria-label="Serbia flag">🇷🇸</span>`
- QR scanner UI: `<button aria-label="Start QR code scanner">` + screen reader announcement when camera opens
- Transaction status icons: CheckCircle, Shield icons have `aria-hidden="true"` + descriptive text nearby
- Decorative SVG patterns: `aria-hidden="true"`

Guideline 1.2: Time-based Media

ID	Criterion	Level	Drop Application
1.2.1	Audio-only and Video-only	A	Not applicable — Drop has no audio/video content
1.2.2	Captions	A	Not applicable
1.2.3	Audio Description or Media Alternative	A	Not applicable
1.2.4	Captions (Live)	AA	Not applicable
1.2.5	Audio Description (Prerecorded)	AA	Not applicable

Future Consideration: If Drop adds tutorial videos or customer support videos, add Norwegian captions + audio descriptions.

Guideline 1.3: Adaptable

ID	Criterion	Level	Drop Application
1.3.1	Info and Relationships	A	Semantic HTML: <code><h1></code> , <code><nav></code> , <code><form></code> , <code><label></code> , <code><button></code> . Tables use <code><table></code> , <code><th></code> , <code><tbody></code>

ID	Criterion	Level	Drop Application
1.3.2	Meaningful Sequence	A	Tab order follows visual order. Reading order is logical (header → main content → navigation)
1.3.3	Sensory Characteristics	A	Instructions do NOT rely on shape/color/sound alone. "Click green button" → "Click 'Send Money' button"
1.3.4	Orientation	AA	No orientation lock. App works in portrait and landscape
1.3.5	Identify Input Purpose	AA	autocomplete attributes on all form fields (email, name, phone, address)

Drop Implementation:

- Login form: `<input type="email" autocomplete="email">`, `<input type="password" autocomplete="current-password">`
- Registration form: `<input autocomplete="given-name">`, `<input autocomplete="family-name">`, `<input autocomplete="tel">`
- Send money form: Amount input has `<label for="amount">Du sender</label>`
- All headings hierarchical: h1 (page title) → h2 (sections) → h3 (subsections)

Guideline 1.4: Distinguishable

ID	Criterion	Level	Drop Application
1.4.1	Use of Color	A	Color is NOT the only way to convey information. Success state = green checkmark icon + "Sendt!" text
1.4.2	Audio Control	A	Not applicable — no auto-playing audio
1.4.3	Contrast (Minimum)	AA	4.5:1 for normal text, 3:1 for large text. Audit all color pairs
1.4.4	Resize Text	AA	Text resizable up to 200% without loss of content or functionality
1.4.5	Images of Text	AA	No images of text except logos. Use web fonts (Inter, Fraunces)
1.4.10	Reflow	AA	Content reflows for 320px viewport without horizontal scrolling

ID	Criterion	Level	Drop Application
1.4.11	Non-text Contrast	AA	3:1 for UI components (buttons, inputs, focus indicators) and graphical objects
1.4.12	Text Spacing	AA	No loss of content when user increases line-height, letter-spacing, word-spacing, paragraph spacing
1.4.13	Content on Hover or Focus	AA	Tooltips dismissible, hoverable, persistent. Not applicable if no tooltips

Drop Color Audit (Priority 1):

Element	Foreground	Background	Contrast Ratio	WCAG AA
Primary button text	#FFFFFF	#0B6E35	TBD	☑ Pass (likely 8:1+)
Body text	#1E293B	#F8FAFC	TBD	Test
Secondary text	#64748B	#F8FAFC	TBD	Test (risky — gray text often fails)
Error text	#EF4444	#FFFFFF	TBD	Test
Link text	#0B6E35	#F8FAFC	TBD	Test
Disabled button	#94A3B8	#F8FAFC	TBD	Exempt (disabled state), but should still aim for 3:1
Focus indicator	#0B6E35	#FFFFFF	TBD	Test (3:1 minimum for non-text)

Tools: Use [WebAIM Contrast Checker](#), axe DevTools, or Lighthouse.

Remediation: If any text fails 4.5:1, darken text color or lighten background. For #64748B (gray), consider #475569 or #334155 (darker grays).

2.2 Principle 2: Operable

User interface components and navigation must be operable.

Guideline 2.1: Keyboard Accessible

ID	Criterion	Level	Drop Application
----	-----------	-------	------------------

2.1.1	Keyboard	A	All functionality available via keyboard (Tab, Enter, Space, Arrow keys)
2.1.2	No Keyboard Trap	A	User can navigate away from any focused component using keyboard alone
2.1.4	Character Key Shortcuts	A	If single-key shortcuts exist, provide way to turn off or remap. Check if any exist.

Drop Implementation:

- **Tab order:** Header → Main content → Bottom navigation → Footer
- **Custom components:** QR scanner button, recipient cards, amount input, confirm button — all keyboard-accessible
- **Modals/dialogs:** Focus trapped inside modal while open, Esc closes modal and returns focus
- **Skip links:** Add "Skip to main content" link (hidden until focused) at top of page
- **Mobile touch targets:** Also keyboard-focusable (44x44px minimum)

Test Plan:

1. Unplug mouse. Navigate entire app using keyboard only.
2. Test all user flows: Login → Dashboard → Send Money → Confirm → Success
3. Check tab order is logical (not jumping around page)
4. Verify focus visible on all interactive elements

Guideline 2.2: Enough Time

ID	Criterion	Level	Drop Application
2.2.1	Timing Adjustable	A	Session timeout warnings with option to extend
2.2.2	Pause, Stop, Hide	A	No auto-updating content except notifications (user can disable in Settings)

Drop Implementation:

- JWT token expires in 1 hour. Show warning 5 minutes before expiry: "Your session will expire in 5 minutes. [Extend session]"
- Dashboard transaction list: NOT auto-refreshing (user clicks Refresh button)
- Notifications: User can toggle push notifications in `/profile/notifications`
- No carousels, no auto-playing animations

Guideline 2.3: Seizures and Physical Reactions

ID	Criterion	Level	Drop Application
2.3.1	Three Flashes or Below Threshold	A	No flashing content. No animations flash more than 3 times per second

Drop Implementation:

- Loading spinners: Slow rotation (1 rotation per 1.5 seconds), no flashing
- Success animation: Fade-in checkmark, no rapid flashing
- No video, no GIF animations

Guideline 2.4: Navigable

ID	Criterion	Level	Drop Application
2.4.1	Bypass Blocks	A	Skip links to bypass repetitive content
2.4.2	Page Titled	A	Every page has unique, descriptive <code><title></code> tag
2.4.3	Focus Order	A	Tab order is logical and predictable
2.4.4	Link Purpose (In Context)	A	Link text describes destination. Avoid "click here" or "read more"
2.4.5	Multiple Ways	AA	More than one way to find a page (menu, search, sitemap)
2.4.6	Headings and Labels	AA	Headings and labels are clear and descriptive
2.4.7	Focus Visible	AA	Keyboard focus indicator visible (outline or border)

Drop Implementation:

- **Skip link:** `<a href="#main-content" class="sr-only focus:not-sr-only">Skip to main content</a>` (Tailwind screen-reader-only class)
- **Page titles:**
 - `/login` → "Logg inn — Drop"
 - `/dashboard` → "Oversikt — Drop"
 - `/send` → "Send penger — Drop"
 - `/transactions` → "Transaksjoner — Drop"
- **Focus indicator:** Default browser outline (blue) or custom 3px solid #0B6E35 outline
- **Navigation:** Bottom nav (Dashboard, Send, Scan, Accounts, Profile) + header back button
- **Breadcrumbs:** Not applicable (flat navigation structure)

- **Link text:** "Opprett konto" (not "Click here to register"), "Se alle transaksjoner" (not "See more")

Guideline 2.5: Input Modalities

ID	Criterion	Level	Drop Application
2.5.1	Pointer Gestures	A	No multipoint or path-based gestures required. Single-tap/click works everywhere
2.5.2	Pointer Cancellation	A	Click/tap activated on up-event (not down-event). Allows user to drag away to cancel
2.5.3	Label in Name	A	Visible label text matches accessible name. Button text = aria-label
2.5.4	Motion Actuation	A	No shake-to-undo or tilt-to-zoom. All features work without device motion

Drop Implementation:

- No swipe gestures required (all actions have buttons)
- QR scanner: Manual "Start Scan" button (no auto-activate on tilt)
- Touch targets: 44x44px minimum (WCAG 2.2 Target Size guideline)

2.3 Principle 3: Understandable

Information and the operation of user interface must be understandable.

Guideline 3.1: Readable

ID	Criterion	Level	Drop Application
3.1.1	Language of Page	A	<code><html lang="nb"></code> (Norwegian Bokmål)
3.1.2	Language of Parts	AA	If mixing languages, mark with <code>lang</code> attribute. Example: <code>Drop</code>

Drop Implementation:

- Primary language: Norwegian Bokmål (`lang="nb"`)
- Brand name "Drop" is English but used as proper noun (no lang tag needed)
- Currency codes (NOK, RSD, BAM) are ISO codes (no lang tag needed)

- Future: If adding English version, separate `/en/` route with `<html lang="en">`

Guideline 3.2: Predictable

ID	Criterion	Level	Drop Application
3.2.1	On Focus	A	Focusing on element does NOT trigger automatic context change
3.2.2	On Input	A	Changing setting does NOT auto-submit form unless user is warned
3.2.3	Consistent Navigation	AA	Navigation in same order on every page
3.2.4	Consistent Identification	AA	Icons/buttons with same function labeled consistently

Drop Implementation:

- No auto-submitting forms. User must click "Bekreft sending" button.
- Bottom nav order consistent: Dashboard → Send → Scan → Accounts → Profile
- Back button (ChevronLeft icon) always top-left corner
- "Send penger" button always labeled "Send penger" (not sometimes "Transfer" or "Pay")

Guideline 3.3: Input Assistance

ID	Criterion	Level	Drop Application
3.3.1	Error Identification	A	Errors clearly identified in text (not color alone)
3.3.2	Labels or Instructions	A	Every form field has label or instruction
3.3.3	Error Suggestion	AA	Suggest fix for errors. "Ugyldig e-postadresse" → show example: "Eksempel: din@epost.no"
3.3.4	Error Prevention (Legal, Financial, Data)	AA	Confirmation step before financial transactions

Drop Implementation:

- **Login errors:** "Feil e-post eller passord" (clear, not vague "Error 401")
- **Registration errors:** "E-post og passord er påkrevd" + field highlighted red border
- **Send money errors:** "Utilstrekkelig saldo. Du har 1,500 NOK tilgjengelig." (specific, actionable)
- **Confirmation step:** Step 3 of 4 shows full transaction details before "Bekreft sending" button

- **Error styling:** Red text (#EF4444) + red border on invalid field + error icon + `aria-invalid="true"` attribute
- **Error announcement:** Use `role="alert"` or `aria-live="polite"` for screen reader announcement

Example Error Component:

```
{error && (  
  <div role="alert" className="rounded-md bg-[#EF4444]/10 p-2 text-sm text-[#EF4444]">  
    {error}  
  </div>  
)}
```

2.4 Principle 4: Robust

Content must be robust enough to be interpreted reliably by a wide variety of user agents, including assistive technologies.

Guideline 4.1: Compatible

ID	Criterion	Level	Drop Application
4.1.1	Parsing	A	Valid HTML (no duplicate IDs, proper nesting, correct ARIA)
4.1.2	Name, Role, Value	A	All UI components have accessible name, role, state (via HTML or ARIA)
4.1.3	Status Messages	AA	Status messages announced to screen readers (via <code>role="status"</code> or <code>aria-live</code>)

Drop Implementation:

- **HTML validation:** Run W3C validator on all pages. Fix parsing errors.
- **ARIA roles:** Radix UI components have built-in ARIA. Custom components need manual ARIA.
- **Status messages:**
 - "Pengene er på vei!" (success) → `<div role="status" aria-live="polite">`
 - "Sender..." (loading) → `<button aria-busy="true">Sender...</button>`
 - "Feil e-post eller passord" (error) → `<div role="alert">`
- **Custom components:**
 - Recipient card button: `<button aria-label="Send penger til {recipient.name}">` (not just icon)

- QR scanner: `<button aria-label="Start QR-skanning">` + `<div role="status">` when camera opens

3. Audit Methodology

3.1 Three-Tier Testing Approach

Tier	Method	Coverage	Tools
Automated	axe-core via Playwright	~30% of WCAG issues	@axe-core/playwright, axe DevTools Chrome extension
Manual	Human testing with keyboard + browser tools	~50% of WCAG issues	Keyboard navigation, browser DevTools, Lighthouse
Assistive Technology	Screen readers on real devices	~20% of WCAG issues	VoiceOver (iOS/macOS), TalkBack (Android), NVDA (Windows)

Why 3 tiers? Automated tools catch low-hanging fruit (missing alt text, color contrast). Manual testing catches logic issues (tab order, focus management). Assistive tech testing catches real-world experience issues (confusing labels, verbose announcements).

3.2 Automated Testing (Tier 1)

Tool: axe-core via Playwright **Integration:** Add accessibility tests to existing Playwright test suite
Frequency: Every PR (CI pipeline), every deployment

Implementation:

1. Install dependencies:

```
npm install --save-dev @axe-core/playwright axe-html-reporter
```

2. Create Playwright accessibility test file:

```
// tests/accessibility.spec.ts
import { test, expect } from '@playwright/test';
import AxeBuilder from '@axe-core/playwright';
import { createHtmlReport } from 'axe-html-reporter';

const routes = [
```

```

    { path: '/', name: 'Landing' },
    { path: '/login', name: 'Login' },
    { path: '/register', name: 'Register' },
    { path: '/dashboard', name: 'Dashboard' },
    { path: '/send', name: 'Send Money' },
    { path: '/scan', name: 'QR Scan' },
    { path: '/accounts', name: 'Bank Accounts' },
    { path: '/transactions', name: 'Transaction History' },
    { path: '/notifications', name: 'Notifications' },
    { path: '/profile', name: 'Profile' },
  ];

  for (const route of routes) {
    test(`${route.name} page should not have accessibility violations`, async ({ page }) => {
      await page.goto(route.path);

      const results = await new AxeBuilder({ page })
        .withTags(['wcag2a', 'wcag2aa', 'wcag21a', 'wcag21aa'])
        .analyze();

      // Generate HTML report
      createHtmlReport({
        results,
        options: {
          outputDir: 'test-results/accessibility',
          reportFileName: `${route.name.toLowerCase().replace(' ', '-')}.html`,
        },
      });

      expect(results.violations).toEqual([]);
    });
  }

```

3. Add to CI pipeline:

```

# .github/workflows/ci.yml
- name: Run accessibility tests
  run: npm run test:ally
- name: Upload accessibility reports

```

```
if: failure()
uses: actions/upload-artifact@v4
with:
  name: accessibility-reports
  path: test-results/accessibility/
```

4. Configure axe rules (optional — custom ruleset):

```
await new AxeBuilder({ page })
  .withTags(['wcag2a', 'wcag2aa', 'wcag21a', 'wcag21aa'])
  .disableRules(['color-contrast']) // Disable if testing on non-production colors
  .analyze();
```

Severity Levels:

- **Critical:** Blocks screen reader users (missing alt text, no keyboard access)
- **Serious:** Major barrier (insufficient contrast, missing labels)
- **Moderate:** Usability issue (redundant links, small touch targets)
- **Minor:** Best practice (missing lang attribute on secondary content)

CI Behavior:

- Fail build on **Critical** violations
- Warn on **Serious** violations (block merge if count increases)
- Report **Moderate/Minor** violations (don't block)

3.3 Manual Testing (Tier 2)

Testers: QA, developers, accessibility specialist **Frequency:** Every major feature, every release candidate

Checklist:

Keyboard Navigation

- ☐ All interactive elements reachable via Tab key
- ☐ Tab order follows visual order (left-to-right, top-to-bottom)
- ☐ Focus indicator visible on all elements (3px outline minimum)
- ☐ No keyboard traps (can navigate away from modals, dropdowns)
- ☐ Enter/Space activates buttons and links
- ☐ Escape closes modals and returns focus to trigger
- ☐ Arrow keys navigate within dropdowns, radio groups (if applicable)

- ☐ Skip link visible when focused (bypasses header to main content)

Form Validation

- ☐ All form fields have visible labels (not just placeholder text)
- ☐ Required fields marked with `required` attribute or `aria-required="true"`
- ☐ Error messages appear near the field (not just at top of form)
- ☐ Error messages announce to screen readers (`role="alert"`)
- ☐ Error messages descriptive ("Ugyldig e-postadresse" not "Error")
- ☐ Invalid fields have `aria-invalid="true"` attribute
- ☐ Success messages announced (`role="status"`)

Color Contrast

- ☐ All text meets 4.5:1 contrast (normal text) or 3:1 (large text ≥ 18 pt)
- ☐ UI components meet 3:1 contrast (buttons, inputs, focus indicators)
- ☐ Links distinguishable from surrounding text (underline or 3:1 contrast)
- ☐ Error/warning/success states not conveyed by color alone (icon + text)

Text Resizing

- ☐ Text resizable to 200% via browser zoom (Cmd/Ctrl + +)
- ☐ No horizontal scrolling at 200% zoom
- ☐ No content cut off or overlapping at 200% zoom
- ☐ Buttons and inputs remain usable at 200% zoom

Responsive Design

- ☐ App works at 320px viewport width (iPhone SE)
- ☐ No horizontal scrolling on mobile
- ☐ Touch targets at least 44x44px (thumb-friendly)
- ☐ No text smaller than 16px on mobile (prevents zoom on iOS)

Headings and Landmarks

- ☐ One `<h1>` per page (page title)
- ☐ Heading hierarchy logical (h1 → h2 → h3, no skipping levels)
- ☐ `<main>` landmark for main content
- ☐ `<nav>` landmark for navigation

☐ `<header>` and `<footer>` landmarks (if applicable)

☐ ARIA landmarks used correctly (not overused)

Tools:

- **Browser DevTools:** Chrome/Edge DevTools → Lighthouse → Accessibility audit
- **Chrome Extensions:** axe DevTools, WAVE, Accessibility Insights
- **Firefox Extensions:** Accessibility Inspector
- **Keyboard only:** Unplug mouse for 30 minutes, test all flows

3.4 Assistive Technology Testing (Tier 3)

Devices:

- **iOS:** VoiceOver on iPhone (iOS 16+)
- **Android:** TalkBack on Pixel or Samsung (Android 12+)
- **macOS:** VoiceOver on MacBook (macOS 13+)
- **Windows:** NVDA (free, open-source) on Windows 10/11

Test Flows:

Flow	Steps	Screen Reader Expectations
Registration	1. Open app → 2. Tap "Opprett konto" → 3. Fill form → 4. Submit	Each field announces label + type + required state. Error messages read aloud. Success confirmation read.
Login	1. Open app → 2. Fill email/password → 3. Submit	Fields announce labels. "Logg inn" button announces "button" role. Error read aloud. Dashboard loads.
Send Money	1. Dashboard → 2. "Send penger" → 3. Select recipient → 4. Enter amount → 5. Confirm	Recipient list announces "list" + number of items. Amount input announces "Du sender" label. Confirmation screen reads all details. Success message announces.
QR Payment	1. Dashboard → 2. "Skann" → 3. Scan QR → 4. Confirm	"Start QR-skanning" button announces. Camera opens + status message "Kamera åpnet". Scanned amount read aloud. Confirmation flow same as Send Money.
Transaction History	1. Dashboard → 2. "Transaksjoner" → 3. Browse list	Transaction list announces "list" + number of items. Each item announces date, amount, recipient. Filter buttons announce state.

Common Issues to Check:

- **Too much information:** Screen reader announces entire page on load (use `aria-live="polite"` sparingly)
- **Too little information:** Button announces "button" but no label (needs `aria-label`)
- **Wrong order:** Visual order $\neq$ DOM order (reorder HTML, don't use CSS to reorder)
- **Unlabeled regions:** "Region" announced without name (add `aria-label` to `<section>`)
- **Redundant announcements:** "Link, link, link" (avoid nested links)
- **Abbreviations:** "NOK" announced as "knock" (use `<abbr title="Norwegian Kroner">NOK</abbr>`)

Testing Protocol:

1. Turn on screen reader
2. Close eyes or look away from screen (simulate blind user)
3. Navigate using screen reader shortcuts only (swipe on mobile, arrow keys on desktop)
4. Can you complete the task without seeing the screen?
5. Are announcements clear and not too verbose?
6. Are interactive elements announced with correct role (button, link, checkbox)?

Documentation: Record screen reader output via screen recording. Share with team to demonstrate issues.

4. Component Audit Plan

Every Drop component and page must be checked for accessibility. Below is the component-by-component audit plan.

4.1 UI Components (Radix UI Primitives)

Drop uses Radix UI, which has **built-in WCAG compliance**. However, **custom styling and usage can break accessibility**, so all components must be audited.

Component	File	WCAG Concerns	Audit Tasks
Button	<code>components/ui/button.tsx</code>	Focus visible, contrast, disabled state	☐ Check focus outline 3:1 contrast, ☐ Check disabled opacity not too low
Input	<code>components/ui/input.tsx</code>	Label association, error state, autocomplete	☐ Check all inputs have <code><label></code> , ☐ Check <code>aria-invalid</code> on errors, ☐ Check <code>autocomplete</code> attributes
Select	<code>components/ui/select.tsx</code>	Keyboard nav, ARIA roles	☐ Test arrow keys, Enter, Esc, ☐ Check <code>aria-expanded</code> , <code>aria-activedescendant</code>

Component	File	WCAG Concerns	Audit Tasks
Dialog	<code>components/ui/dialog.tsx</code>	Focus trap, Esc close, focus return	☐ Tab stays inside modal, ☐ Esc closes and returns focus, ☐ Check <code>aria-labelledby</code>
Alert	<code>components/ui/alert.tsx</code>	Screen reader announcement	☐ Check <code>role="alert"</code> or <code>aria-live="assertive"</code> for critical alerts
Card	<code>components/ui/card.tsx</code>	Semantic structure	☐ Check not used as button (use <code><button></code> wrapper if clickable)
Tabs	<code>components/ui/tabs.tsx</code>	ARIA roles, keyboard nav	☐ Check <code>role="tablist"</code> , <code>role="tab"</code> , <code>role="tabpanel"</code> , ☐ Arrow keys switch tabs
Sheet	<code>components/ui/sheet.tsx</code>	Focus trap, Esc close	Same as Dialog
Badge	<code>components/ui/badge.tsx</code>	Not interactive	☐ Verify NOT focusable (decorative only)
Skeleton	<code>components/ui/skeleton.tsx</code>	Loading state announcement	☐ Check <code>aria-busy="true"</code> or <code>role="status"</code> with "Laster..." text

Radix UI Accessibility Features (Built-in):

- Focus management (focus trap in modals, focus return on close)
- Keyboard navigation (Arrow keys, Enter, Space, Esc)
- ARIA attributes (`role`, `aria-expanded`, `aria-selected`, `aria-labelledby`, `aria-describedby`)
- Screen reader announcements (live regions for dynamic content)

Reference: [Radix UI Accessibility Docs](#)

4.2 Custom Components

Component	File	WCAG Concerns	Audit Tasks
DropLogoFull	<code>components/drop-logo.tsx</code>	Alt text	☐ Check <code><svg role="img" aria-label="Drop logo"></code> or <code></code>
BottomNav	<code>components/bottom-nav.tsx</code>	ARIA roles, active state	☐ Check <code><nav role="navigation"></code> , ☐ Active tab has <code>aria-current="page"</code> , ☐ Icons have labels

Component	File	WCAG Concerns	Audit Tasks
QRScanner	<code>components/qr-scanner.tsx</code>	Camera access, error handling	- Button labeled "Start QR-skanning", - Camera error announced, - Scanned value announced
AuthProvider	<code>components/auth-provider.tsx</code>	Loading state	Check loading spinner has <code>aria-label="Laster brukerdata"</code>
ErrorBoundary	<code>components/error-boundary.tsx</code>	Error announcement	- Error message has <code>role="alert"</code>, - Retry button keyboard-accessible
CookieConsent	<code>components/cookie-consent.tsx</code>	Focus management	- Focus moves to consent dialog on page load, - Buttons keyboard-accessible
PrePaymentDisclosure	<code>components/pre-payment-disclosure.tsx</code>	Readability	- Text 4.5:1 contrast, - Links clearly labeled
DropIcons	<code>components/drop-icons.tsx</code>	Decorative vs informative	- Decorative icons have <code>aria-hidden="true"</code>, - Informative icons have <code>aria-label</code>

4.3 Pages (Routes)

Page	Route	WCAG Concerns	Audit Tasks
Landing	<code>/</code>	Headings, link labels	- h1 "Drop", "Opprett konto" and "Logg inn" buttons clear, - Skip link
Login	<code>/login</code>	Form labels, error handling	- Email/password labeled, - Errors announced, - BankID button keyboard-accessible
Register	<code>/register</code>	Form labels, validation	- All fields labeled, - Age validation error clear, - Phone format error shows example
Onboarding	<code>/onboarding</code>	Multi-step form	- Progress indicator has <code>aria-label="Steg 1 av 3"</code>, "Neste" and "Tilbake" buttons clear
Dashboard	<code>/dashboard</code>	Headings, transaction list	- h1 "Oversikt", - Transaction list has semantic <code></code>, - Balance announced clearly

Page	Route	WCAG Concerns	Audit Tasks
Send Money	<code>/send</code>	Multi-step flow, confirmation	- Each step has h1, - Recipient list keyboard-navigable, - Amount input labeled, - Confirmation summary read aloud
QR Scan	<code>/scan</code>	Camera access, error handling	- Permission error announced, - Manual entry alternative available, - Camera button labeled
Bank Accounts	<code>/accounts</code>	Account list, sync status	- Account list semantic, "Synkroniser" button labeled, - Last synced time announced
Transaction History	<code>/transactions</code>	Filters, date range	- Filter buttons keyboard-accessible, - Date picker keyboard-navigable, - Transaction count announced
Notifications	<code>/notifications</code>	List, mark as read	- Notification list semantic, - Unread badge announced, "Merk som lest" button labeled
Profile	<code>/profile</code>	Settings, logout	- Settings sections have h2, - Toggle switches have labels, - Logout button confirmation
Privacy	<code>/privacy</code>	Long text, readability	- Headings hierarchical, - Paragraphs not too wide (60-80 chars), - Line height 1.5+
Terms	<code>/terms</code>	Long text, readability	Same as Privacy
Fees	<code>/fees</code>	Pricing table	- Table has <code><th></code>, <code><caption></code>, - Currency amounts clearly labeled

5. Critical User Flows — Accessibility Requirements

Financial transactions require extra-clear confirmation states and error handling for accessibility.

5.1 Registration Flow

Steps: Landing → Register → Onboarding → BankID Verification → Dashboard

Accessibility Requirements:

Step	Requirement	WCAG Criterion
Registration form	All fields labeled with <code><label for="..."></code>	3.3.2
Email validation	Error message specific: "Ugyldig e-postadresse. Eksempel: din@epost.no"	3.3.1, 3.3.3
Password requirements	Requirements shown near password field (not just in tooltip)	3.3.2
Age validation	Error if < 18: "Du må være minst 18 år for å bruke Drop"	3.3.1
Phone validation	Error if not +47: "Kun norske telefonnumre (+47) er tillatt"	3.3.1
BankID redirect	Button labeled "Bekreft med BankID" + description "Du vil bli sendt til BankID"	2.4.4
Loading state	"Verifiserer med BankID..." with <code>aria-busy="true"</code>	4.1.2
Success	"Konto opprettet!" with <code>role="status"</code> + auto-redirect to dashboard in 3 seconds	4.1.3

Test with screen reader: Does user understand each step? Are errors clear? Is success confirmed?

5.2 Login Flow

Steps: Landing → Login → Dashboard

Accessibility Requirements:

Step	Requirement	WCAG Criterion
Email field	<code><label for="email">E-postadresse</label></code> + <code>autocomplete="email"</code>	1.3.5, 3.3.2
Password field	<code><label for="password">Passord</label></code> + <code>autocomplete="current-password"</code> + show/hide toggle	1.3.5, 3.3.2
Error	"Feil e-post eller passord" with <code>role="alert"</code> + focus moves to email field	3.3.1

Step	Requirement	WCAG Criterion
Loading	"Logger inn..." with <code>aria-busy="true"</code> on button	4.1.2
Success	Auto-redirect to dashboard (no announcement needed)	N/A
BankID option	"Logg inn med BankID" button keyboard-accessible + icon has <code>aria-hidden="true"</code>	2.1.1

Test with screen reader: Can user complete login without seeing screen? Are errors clear?

5.3 Send Money Flow

Steps: Dashboard → Send → Select Recipient → Enter Amount → Confirm → Success

Accessibility Requirements:

Step	Requirement	WCAG Criterion
Progress indicator	"Steg 1 av 4" with <code>aria-label</code> + visual progress bar (not color alone)	1.4.1
Recipient list	<code><ul role="list"></code> with <code>aria-label="Mottakere"</code> + each item keyboard-focusable	4.1.2
Search field	<code><label for="search">Søk mottakere</label></code> + clear button labeled	3.3.2
No results	"Ingen mottakere funnet" announced via <code>role="status"</code>	4.1.3
Amount input	<code><label for="amount">Du sender</label></code> + currency shown as text (not just symbol)	1.3.1, 3.3.2
Exchange rate	"1 NOK = 10.5 RSD" read aloud (not just visual)	1.3.1
Fee calculation	"Gebyr: 5.00 NOK (0.5%)" announced when amount changes	4.1.3
Total	"Totalt: 1,005.00 NOK" with <code>aria-live="polite"</code>	4.1.3
Confirmation screen	All details read aloud: amount, recipient, fee, total, delivery time	1.3.1
Confirm button	"Bekreft sending" button with focus on load + Enter activates	2.4.7
Loading	"Sender..." with <code>aria-busy="true"</code> on button + spinner <code>aria-hidden="true"</code>	4.1.2

Step	Requirement	WCAG Criterion
Error	"Utilstrekkelig saldo. Du har 1,500 NOK tilgjengelig." with <code>role="alert"</code>	3.3.1, 3.3.3
Success	"Pengene er på vei!" with <code>role="status"</code> + checkmark icon <code>aria-hidden="true"</code> + success sound (optional, user can disable)	4.1.3

Test with screen reader: Can blind user send money confidently? Are amounts clear? Is confirmation explicit?

Test with keyboard only: Can user complete flow without mouse? Is tab order logical?

5.4 QR Payment Flow

Steps: Dashboard → Scan → Camera Access → Scan QR → Confirm → Success

Accessibility Requirements:

Step	Requirement	WCAG Criterion
Start scan button	"Start QR-skanning" with icon <code>aria-hidden="true"</code>	1.1.1, 2.4.4
Camera permission	If denied: "Kamera-tilgang nektet. Aktiver kamera i innstillinger." with <code>role="alert"</code>	3.3.1
Camera active	"Kamera åpnet. Hold QR-koden foran kameraet." with <code>role="status"</code>	4.1.3
Manual entry option	"Eller skriv inn beløp manuelt" link visible + keyboard-accessible	2.4.1, 2.4.4
Scanned value	"Skannet: 150 NOK til Rema 1000" announced via <code>role="status"</code>	4.1.3
Confirmation	Same as Send Money flow	3.3.4

Accessibility Challenge: QR scanning is inherently visual. How do blind users pay merchants?

Solution:

- 1. **Manual entry option:** Merchant shows amount on screen, blind user asks sighted person or merchant to read amount, user enters manually.
- 2. **NFC tap-to-pay (future):** Accessible alternative to QR codes (works with screen curtain on iPhone).
- 3. **Audio QR codes (future):** QR code embeds tone sequence, app decodes via microphone.

Test with screen reader: Can blind user understand QR scanner is active? Is manual entry alternative discoverable?

5.5 Transaction History Flow

Steps: Dashboard → Transaksjoner → Filter → View Details

Accessibility Requirements:

Step	Requirement	WCAG Criterion
Transaction list	<code><ul role="list" aria-label="Transaksjoner"></code> + count announced: "20 transaksjoner"	4.1.2
Each item	"15. februar, Sent til Marko, 500 NOK" (date, action, recipient, amount)	1.3.1
Filter buttons	"Filter: Alle" (default) with <code>aria-pressed="true"</code> on active filter	4.1.2
Date range picker	Keyboard-navigable calendar with arrow keys	2.1.1
No results	"Ingen transaksjoner funnet" with <code>role="status"</code>	4.1.3
Load more	"Last flere transaksjoner" button at bottom (not infinite scroll)	2.2.2

Test with screen reader: Can user browse transaction history? Are filters clear? Is date range picker usable?

6. Testing Tools

6.1 Automated Tools

Tool	Purpose	Cost	Integration
@axe-core/playwright	WCAG automated testing in CI	Free (open-source)	Playwright test suite
axe DevTools (Chrome)	Manual audit during development	Free tier available	Browser extension
Lighthouse	Overall accessibility score + performance	Free (built into Chrome)	Chrome DevTools

Tool	Purpose	Cost	Integration
WAVE	Visual feedback on accessibility issues	Free	Browser extension
Pa11y	Command-line accessibility testing	Free (open-source)	CI pipeline (alternative to axe)

6.2 Manual Testing Tools

Tool	Purpose	Platform	Cost
VoiceOver	Screen reader	iOS, macOS	Free (built-in)
TalkBack	Screen reader	Android	Free (built-in)
NVDA	Screen reader	Windows	Free (open-source)
JAWS	Screen reader	Windows	\$1,000+ (enterprise, most popular)
Accessibility Insights	Guided manual tests	Chrome, Edge	Free (Microsoft)
Color Contrast Analyzer	Check contrast ratios	Desktop app	Free (TPGi)
HeadingsMap	Visualize heading structure	Browser extension	Free

6.3 Testing Checklist Template

Use this checklist for each release:

```
## Accessibility Audit – [Feature Name] – [Date]

### Automated Testing
- [ ] axe-core tests pass (0 critical violations)
- [ ] Lighthouse accessibility score ≥ 90
- [ ] WAVE reports no errors

### Manual Testing
- [ ] Keyboard navigation works (no traps, logical order)
- [ ] Focus indicators visible (3:1 contrast)
- [ ] Color contrast meets WCAG AA (4.5:1 text, 3:1 UI)
- [ ] Form labels present and associated
- [ ] Error messages clear and actionable
- [ ] Success messages announced to screen readers
- [ ] Page titles unique and descriptive
- [ ] Headings hierarchical (h1 → h2 → h3)
```

- [] Text resizable to 200% without scrolling

Screen Reader Testing

- [] VoiceOver (iOS) – All flows completable
- [] TalkBack (Android) – All flows completable
- [] NVDA (Windows) – All flows completable
- [] Announcements clear and not too verbose

Critical Violations

[List any critical violations found]

Remediation Plan

[List fixes needed before launch]

Sign-off

- [] QA Lead
- [] Accessibility Specialist (if available)
- [] Product Owner

7. Remediation Priority Framework

Not all accessibility issues are equal. Prioritize fixes based on **impact** (how many users affected) and **severity** (how badly affected).

7.1 Priority Matrix

Priority	Severity	Impact	Example	Timeline
P0 — Critical	Blocker	High	Missing alt text on critical images, form without labels, keyboard trap in payment flow	Fix before launch
P1 — High	Major barrier	Medium-High	Insufficient color contrast on body text, missing error messages, broken screen reader announcements	Fix in current sprint

Priority	Severity	Impact	Example	Timeline
P2 — Medium	Usability issue	Medium	Redundant link text, missing skip link, small touch targets (< 44px)	Fix in next sprint
P3 — Low	Best practice	Low	Missing lang attribute on secondary content, verbose ARIA labels	Fix when time allows

7.2 Violation Examples by Priority

P0 — Critical (Fix immediately)

- Login form has no labels → Screen reader users cannot log in
- Payment confirmation button not keyboard-accessible → Keyboard users cannot send money
- Error messages shown only as red border (no text) → Blind users don't know what's wrong
- Modal dialog traps focus (cannot Esc close) → Keyboard users stuck

P1 — High (Fix this sprint)

- Body text (#64748B on #F8FAFC) has 3.2:1 contrast (fails WCAG AA 4.5:1)
- "Click here" link text (destination unclear)
- Transaction list missing semantic structure (`<div>` instead of `<ul>`)
- Loading spinner has no announcement (user doesn't know page is loading)

P2 — Medium (Fix next sprint)

- Touch target is 40x40px (should be 44x44px)
- Heading hierarchy skips level (h1 → h3)
- Page title not unique ("Drop" on every page)
- Image has alt text but it's too verbose ("Green rounded square logo with white dollar icon circular arrows and gold dot top-right")

P3 — Low (Fix when time allows)

- ARIA label redundant with visible text (`<button aria-label="Send penger">Send penger</button>`)
- Missing `lang="en"` on "Drop" brand name
- Abbreviation not expanded (`NOK` should be `<abbr title="Norske kroner">NOK</abbr>`)

7.3 Triage Process

1. **Run automated tests** → Generate violation report

2. **Classify violations** → Assign P0/P1/P2/P3 priority
 3. **Create tasks** → P0/P1 go into current sprint backlog
 4. **Assign owners** → Developer + QA pair for each P0/P1 issue
 5. **Verify fixes** → Re-run automated tests + manual testing
 6. **Document** → Update accessibility report
-

8. CI Integration

Accessibility tests must run on every PR and deployment to prevent regressions.

8.1 GitHub Actions Workflow

Add accessibility testing step to existing CI pipeline:

```
# .github/workflows/ci.yml
name: CI

on:
  pull_request:
  push:
    branches: [main, staging]

jobs:
  accessibility:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4

      - name: Setup Node
        uses: actions/setup-node@v4
        with:
          node-version: '20'
          cache: 'npm'

      - name: Install dependencies
        run: npm ci

      - name: Build app
```

```

run: npm run build

- name: Run accessibility tests
  run: npm run test:ally

- name: Upload accessibility reports
  if: failure()
  uses: actions/upload-artifact@v4
  with:
    name: accessibility-reports
    path: test-results/accessibility/
    retention-days: 7

- name: Comment PR with violations
  if: failure() && github.event_name == 'pull_request'
  uses: actions/github-script@v7
  with:
    script: |
      const fs = require('fs');
      const report = fs.readFileSync('test-results/accessibility/summary.txt', 'utf8');
      github.rest.issues.createComment({
        issue_number: context.issue.number,
        owner: context.repo.owner,
        repo: context.repo.repo,
        body: `## ☐ Accessibility Violations Found\n\n${report}\n\nSee full report in
artifacts.`
      });

```

8.2 Fail Conditions

Block merge if:

- ≥ 1 **Critical** violation (blocking issue)
- ≥ 5 **Serious** violations (major accessibility barriers)

Warn but allow merge if:

- < 5 **Serious** violations (review required)
- Any number of **Moderate/Minor** violations (fix later)

Implementation:

```
// tests/accessibility.spec.ts
const results = await new AxeBuilder({ page }).analyze();

const critical = results.violations.filter(v => v.impact === 'critical');
const serious = results.violations.filter(v => v.impact === 'serious');

if (critical.length > 0) {
  throw new Error(`CRITICAL: ${critical.length} critical accessibility violations found. Merge
blocked.`);
}

if (serious.length >= 5) {
  throw new Error(`SERIOUS: ${serious.length} serious accessibility violations found. Merge
blocked.`);
}

// Warn for moderate/minor
if (results.violations.length > 0) {
  console.warn(`⚠️ ${results.violations.length} accessibility violations found (non-
blocking)`);
}
```

8.3 Pre-commit Hook (Optional)

Prevent developers from committing inaccessible code:

```
# .husky/pre-commit
#!/bin/sh
npm run lint
npm run test:ally:quick # Fast subset of tests (login, dashboard only)
```

9. Accessibility Reporting Template

After each audit, generate a report for stakeholders.

9.1 Report Structure

Drop Accessibility Audit Report

****Date:**** [YYYY-MM-DD]
****Audited by:**** [Name]
****Scope:**** [All pages / Specific feature]
****Standard:**** WCAG 2.1 Level AA

Executive Summary

- ****Overall Score:**** [Lighthouse accessibility score] / 100
- ****Critical Violations:**** [Number]
- ****Serious Violations:**** [Number]
- ****Moderate Violations:**** [Number]
- ****Minor Violations:**** [Number]
- ****Compliance Status:**** ☐ Compliant / ☒ Non-compliant (remediation in progress) / ☐ Non-compliant (action required)

Detailed Findings

Critical Violations (P0)

ID	Page	Issue	WCAG Criterion	Affected Users	Fix
1	/login	Email input missing label	3.3.2	Screen reader users	Add <code><label for="email">E-postadresse</label></code>
2	/send	Confirm button not keyboard-accessible	2.1.1	Keyboard users	Change <code><div onClick></code> to <code><button></code>

Serious Violations (P1)

[Same table format]

Moderate Violations (P2)

[Same table format]

Minor Violations (P3)

[Same table format]

Compliance by WCAG Principle

Principle	Criteria Tested			Pass	Fail	Pass Rate
Perceivable	25	22	3	88%		
Operable	20	18	2	90%		
Understandable	12	12	0	100%		
Robust	3	3	0	100%		
Total	60	55	5	92%		

Remediation Plan

Priority	Issue Count	Assigned To	Deadline
P0 (Critical)	2	Dev Team	Before launch
P1 (High)	3	Dev Team	Sprint 24
P2 (Medium)	8	Dev Team	Sprint 25-26
P3 (Low)	12	Backlog	TBD

Testing Methodology

- **Automated:** axe-core via Playwright (30% coverage)
- **Manual:** Keyboard navigation, color contrast, form validation (50% coverage)
- **Assistive Technology:** VoiceOver (iOS), TalkBack (Android), NVDA (Windows) (20% coverage)

Sign-off

- [] QA Lead: [Name]

- [] Accessibility Specialist: [Name]
- [] Product Owner: [Name]
- [] CEO: Alem (final approval before launch)

10. Compliance Statement (Tilgjengelighetserklæring)

Norwegian law requires a public accessibility statement (tilgjengelighetserklæring) on the website.

10.1 Required Content

- **Compliance level:** WCAG 2.1 Level AA
- **Date of last audit:** [Date]
- **Known issues:** List any non-compliant features (if any)
- **Contact for accessibility issues:** Email or form
- **Complaint process:** How to file a complaint with Digitaliseringsdirektoratet

10.2 Template (Norwegian)

Create page at `/tilgjengelighet`:

Tilgjengelighetserklæring for Drop

****Sist oppdatert:**** [YYYY-MM-DD]

Drop forplikter seg til å gjøre våre tjenester tilgjengelige for alle, inkludert personer med funksjonsnedsettelse. Denne erklæringen beskriver i hvilken grad Drop oppfyller kravene til universell utforming av IKT-løsninger.

Overholdelse av WCAG 2.1

Drop sikter mot å oppfylle ****WCAG 2.1 nivå AA**** i henhold til forskrift om universell utforming av IKT-løsninger.

Kjente begrensninger

Per [dato] har vi identifisert følgende områder som ikke fullt ut oppfyller WCAG 2.1 AA:

- [Liste eventuelle kjente problemer, eller skriv "Ingen kjente begrensninger"]

Vi arbeider kontinuerlig med å forbedre tilgjengeligheten til våre tjenester.

Tilbakemelding

Har du opplevd problemer med tilgjengeligheten til Drop? Vi vil gjerne høre fra deg.

Kontakt: support@getdrop.no

Beskriv problemet så detaljert som mulig, inkludert:

- Hvilken side eller funksjon du prøvde å bruke
- Hvilken enhet og nettleser du brukte
- Eventuell hjelpeteknologi du brukte (f.eks. skjermleser)

Vi vil svare på henvendelser om tilgjengelighet innen 5 virkedager.

Klageadgang

Dersom du ikke er fornøyd med vårt svar, kan du klage til Digitaliseringsdirektoratet:

Digitaliseringsdirektoratet

Postboks 1382 Vika, 0114 Oslo

E-post: post@digdir.no

Nettside: <https://www.digdir.no/>

Revisjon av erklæringen

Denne erklæringen ble opprettet [dato] og sist oppdatert [dato]. Vi gjennomgår og oppdaterer denne erklæringen minst én gang per år.

10.3 English Version

Also provide English version at </en/accessibility> for international users.

11. Training & Developer Guidelines

Accessibility is everyone's responsibility. All developers must understand basic WCAG principles.

11.1 Developer Onboarding Checklist

New developers must complete:

- ☐ Read this accessibility spec (30 min)
- ☐ Watch "Introduction to Screen Readers" video (20 min)
- ☐ Install axe DevTools Chrome extension
- ☐ Run accessibility audit on 1 page (hands-on exercise)
- ☐ Fix 1 accessibility bug (pair with senior dev)

11.2 Coding Guidelines (Quick Reference)

Semantic HTML First

```
// ❌ Bad
<div onClick={handleClick}>Click me</div>

// ✅ Good
<button onClick={handleClick}>Click me</button>
```

Always Label Inputs

```
// ❌ Bad
<input type="email" placeholder="Email" />

// ✅ Good
<label htmlFor="email">E-postadresse</label>
<input id="email" type="email" autocomplete="email" />
```

Focus Indicators

```
/* ❌ Bad */
button:focus {
  outline: none; /* NEVER do this without custom focus style */
}

/* ✅ Good */
button:focus-visible {
```

```
outline: 3px solid #0B6E35;
outline-offset: 2px;
}
```

Error Announcements

```
// ❌ Bad
{error && <p className="text-red-500">{error}</p>}

// ✅ Good
{error && <p role="alert" className="text-red-500">{error}</p>}
```

Loading States

```
// ❌ Bad
<button disabled={loading}>
  {loading ? <Spinner /> : "Send"}
</button>

// ✅ Good
<button disabled={loading} aria-busy={loading}>
  {loading ? "Sender..." : "Send"}
</button>
```

Images

```
// ❌ Bad (decorative icon treated as informative)


// ✅ Good (decorative)


// ✅ Good (informative)

```

11.3 Code Review Checklist

Every PR must pass this accessibility checklist:

- ☐ All interactive elements keyboard-accessible (Tab, Enter, Space)

- ☐ All form inputs have labels (visible or `aria-label`)
- ☐ Color contrast meets WCAG AA (4.5:1 text, 3:1 UI)
- ☐ Error messages have `role="alert"`
- ☐ Success messages have `role="status"`
- ☐ Images have alt text (or `alt=""` if decorative)
- ☐ Buttons/links have descriptive text (not "Click here")
- ☐ Focus indicators visible on all interactive elements
- ☐ No `outline: none` without custom focus style
- ☐ Page has unique `<title>` tag
- ☐ Headings hierarchical (h1 → h2 → h3)
- ☐ ARIA used correctly (not overused or misused)

12. Implementation Plan (Phased Approach)

Full accessibility compliance is a multi-sprint effort. Below is a phased rollout plan.

Phase 1: Foundation (Sprint 1-2) — Before MVP Launch

Goal: Fix all P0 (critical) violations. Minimum viable accessibility.

Task	Owner	Acceptance Criteria
Add labels to all form inputs	Dev Team	All inputs have <code><label></code> or <code>aria-label</code>
Fix keyboard navigation	Dev Team	All pages navigable via Tab key
Add focus indicators	Dev Team	All interactive elements have visible focus (3px outline)
Fix critical color contrast	Design + Dev	Body text meets 4.5:1, buttons meet 3:1
Add error announcements	Dev Team	All errors have <code>role="alert"</code>
Add page titles	Dev Team	All pages have unique <code><title></code>
Set up axe-core CI tests	DevOps	CI fails on critical violations
Create accessibility statement	Content Team	<code>/tilgjengelighet</code> page published

Deliverable: Lighthouse accessibility score ≥ 80, 0 critical violations

Phase 2: Compliance (Sprint 3-4) — Pre-Public Launch

Goal: Full WCAG 2.1 AA compliance. Pass Digdir audit.

Task	Owner	Acceptance Criteria
Fix all P1 (high) violations	Dev Team	Serious violations < 5
Screen reader testing (iOS, Android, Windows)	QA + Accessibility Specialist	All critical flows completable
Add skip links	Dev Team	Skip to main content on all pages
Improve heading hierarchy	Dev Team	All pages have h1, logical h2/h3 structure
Add ARIA landmarks	Dev Team	<main>, <nav>, <header>, <footer> on all pages
Fix text resizing issues	Dev Team	Content usable at 200% zoom
Add autocomplete attributes	Dev Team	All form fields have autocomplete
Manual keyboard testing (all flows)	QA	No keyboard traps, tab order logical
Full accessibility audit report	Accessibility Specialist	Report delivered to CEO

Deliverable: Lighthouse accessibility score ≥ 90, WCAG 2.1 AA compliant

Phase 3: Optimization (Sprint 5-6) — Post-Launch

Goal: Exceed WCAG 2.1 AA. Best-in-class accessibility for fintech.

Task	Owner	Acceptance Criteria
Fix all P2 (medium) violations	Dev Team	Moderate violations < 10
Add alternative text improvements	Content Team	All alt text concise and descriptive
Improve ARIA label clarity	Dev Team	Labels tested with real screen reader users
Add QR code alternatives	Dev Team	Manual entry option visible and easy to find
Improve transaction announcements	Dev Team	Screen reader announces amounts clearly

Task	Owner	Acceptance Criteria
Add accessibility preference controls	Dev Team	User can reduce motion, increase contrast
Conduct user testing with disabled users	Research Team	5+ users with disabilities test app
WCAG 2.2 upgrade (optional)	Dev Team	Evaluate new success criteria (Target Size 2.5.8, Focus Not Obscured 2.4.11)

Deliverable: Lighthouse accessibility score ≥ 95 , user testimonials from disabled users

13. Budget & Resources

13.1 Estimated Effort

Phase	Effort (Dev Days)	Timeline
Phase 1: Foundation	10-15 days	Sprint 1-2 (2 weeks)
Phase 2: Compliance	15-20 days	Sprint 3-4 (2 weeks)
Phase 3: Optimization	10-15 days	Sprint 5-6 (2 weeks)
Total	35-50 days	6 weeks

Assumptions:

- 2 full-time developers (50% time on accessibility)
- 1 QA engineer (25% time on accessibility testing)
- 1 accessibility specialist (contractor, 20 hours consultation)

13.2 Tool Costs

Tool	Cost	Justification
axe-core (open-source)	Free	CI integration, automated testing
Lighthouse (built into Chrome)	Free	Manual testing, audit reports
NVDA (open-source)	Free	Windows screen reader testing
VoiceOver (built into iOS/macOS)	Free	Apple screen reader testing
TalkBack (built into Android)	Free	Android screen reader testing
Accessibility Insights (Microsoft)	Free	Guided manual testing

Tool	Cost	Justification
Optional: Accessibility Specialist	~\$2,000 USD (20 hours @ \$100/hr)	External audit, training, consultation
Total	\$0 - \$2,000	Optional contractor cost

13.3 ROI (Return on Investment)

Compliance Benefits:

- **Legal:** Avoid fines from Digitaliseringsdirektoratet
- **Market:** 15% of population has disabilities — expand addressable market
- **Reputation:** First fully accessible fintech in Norway → competitive advantage
- **SEO:** Better semantic HTML → better Google rankings
- **Usability:** Accessible design benefits all users (clear labels, logical flow, keyboard shortcuts)

Example: If Drop has 10,000 users, 15% (1,500) have some disability. If accessibility unlocks this segment, revenue increase = 15%.

Appendix A: WCAG 2.1 AA Complete Criteria List

Full list of all 50 WCAG 2.1 Level A/AA success criteria:

Level A (25 criteria)

Perceivable:

- 1.1.1 Non-text Content
- 1.2.1 Audio-only and Video-only (Prerecorded)
- 1.2.2 Captions (Prerecorded)
- 1.2.3 Audio Description or Media Alternative (Prerecorded)
- 1.3.1 Info and Relationships
- 1.3.2 Meaningful Sequence
- 1.3.3 Sensory Characteristics
- 1.4.1 Use of Color
- 1.4.2 Audio Control

Operable:

- 2.1.1 Keyboard

- 2.1.2 No Keyboard Trap
- 2.1.4 Character Key Shortcuts
- 2.2.1 Timing Adjustable
- 2.2.2 Pause, Stop, Hide
- 2.3.1 Three Flashes or Below Threshold
- 2.4.1 Bypass Blocks
- 2.4.2 Page Titled
- 2.4.3 Focus Order
- 2.4.4 Link Purpose (In Context)
- 2.5.1 Pointer Gestures
- 2.5.2 Pointer Cancellation
- 2.5.3 Label in Name
- 2.5.4 Motion Actuation

Understandable:

- 3.1.1 Language of Page
- 3.2.1 On Focus
- 3.2.2 On Input
- 3.3.1 Error Identification
- 3.3.2 Labels or Instructions

Robust:

- 4.1.1 Parsing
- 4.1.2 Name, Role, Value

Level AA (25 additional criteria)

Perceivable:

- 1.2.4 Captions (Live)
- 1.2.5 Audio Description (Prerecorded)
- 1.3.4 Orientation
- 1.3.5 Identify Input Purpose
- 1.4.3 Contrast (Minimum)
- 1.4.4 Resize Text
- 1.4.5 Images of Text
- 1.4.10 Reflow
- 1.4.11 Non-text Contrast
- 1.4.12 Text Spacing
- 1.4.13 Content on Hover or Focus

Operable:

- 2.4.5 Multiple Ways

- 2.4.6 Headings and Labels
- 2.4.7 Focus Visible

Understandable:

- 3.1.2 Language of Parts
- 3.2.3 Consistent Navigation
- 3.2.4 Consistent Identification
- 3.3.3 Error Suggestion
- 3.3.4 Error Prevention (Legal, Financial, Data)

Robust:

- 4.1.3 Status Messages
-

Appendix B: Norwegian Accessibility Law Summary

Primary Law: Likestillings- og diskrimineringsloven (Equality and Anti-Discrimination Act)

Key Section: § 18 — Duty to provide universal design

Scope: Public and private sector digital services (including fintech, e-commerce, education)

Standard: WCAG 2.1 Level AA (minimum)

Enforcement: Digitaliseringsdirektoratet (Norwegian Digitalisation Agency)

Penalties: Fines for non-compliance, discrimination complaints to LDO (Equality and Anti-Discrimination Ombud)

Deadline: Norway universally designed by 2025 (national goal)

Drop's Obligation: As a fintech service provider operating in Norway, Drop MUST comply with universal design requirements before public launch. Failure to comply risks fines, legal liability, and reputational damage.

Document Control

Version History:

Version	Date	Author	Changes
1.0	2026-02-17	Architect Agent	Initial draft — full WCAG 2.1 AA spec + Norwegian law + audit methodology

Approvals:

- ☒ Architect Agent — 2026-02-17
- ☐ John (AI Director) — Pending
- ☐ Alem (CEO) — Pending

Next Steps:

- 1. Review by John (validate Norwegian law references, tool recommendations)
- 2. Approval by Alem (finalize budget and timeline)
- 3. Share with Dev Team (begin Phase 1 implementation)
- 4. Log to HiveMind (post intel entry)

END OF SPECIFICATION

Revision #5
Created 2026-02-18 08:44:43 UTC by John
Updated 2026-07-12 20:00:26 UTC by John