

Epic 4.1: Navigation Organisms

Epic Overview

This epic creates sophisticated navigation organism components that provide consistent, accessible, and workspace-aware navigation experiences across THE WHEEL design system applications.

Priority: P0 (Critical)

Timeline: 3 weeks

Dependencies: Epic 3.3 (Interactive Molecules), Epic 2.4 (Icon & Asset System)

Story 4.1.1: Primary Navigation

Overview

Create comprehensive primary navigation organisms including top navigation, side navigation, and mobile navigation with full workspace context awareness.

AI Developer Prompt

You are enhancing primary navigation organisms for THE WHEEL design system. Building on the molecule components from Feature 3, you need to create sophisticated navigation systems that provide consistent workspace navigation experiences.

Context

- Complete molecule component system with workspace context
- Existing TopNavigation, SideNavigation, MobileNav components
- Need workspace-aware navigation with context switching
- Navigation is critical for workspace application usability
- Must support responsive design and accessibility

Requirements

1. Enhance TopNavigation with workspace context:

- Workspace switcher integration
- Context-aware navigation items
- User profile and settings integration
- Notification center integration

- Responsive navigation behavior

2. Enhance SideNavigation with workspace navigation:

- Workspace-specific navigation hierarchies
- Collapsible navigation sections
- Active state management
- Permission-based navigation items
- Responsive sidebar behavior

3. Enhance MobileNav with workspace features:

- Mobile-optimized workspace switching
- Touch-friendly navigation interactions
- Swipe gestures for navigation
- Bottom sheet navigation patterns
- Accessibility improvements for mobile

Specific Tasks

- ☐ Enhance TopNavigation with workspace context
- ☐ Add workspace switcher integration
- ☐ Enhance SideNavigation with workspace navigation
- ☐ Add collapsible navigation sections
- ☐ Enhance MobileNav with workspace features
- ☐ Implement responsive navigation behavior
- ☐ Add comprehensive accessibility features

Documentation Required

- Navigation system architecture
- Workspace navigation patterns
- Responsive navigation implementation
- Accessibility best practices
- Navigation state management
- Mobile navigation guidelines

Testing Requirements

- Navigation functionality tests
- Workspace context switching tests
- Responsive navigation tests
- Accessibility compliance tests
- Mobile navigation tests
- Permission-based navigation tests
- Cross-browser compatibility tests

Integration Points

- Integration with workspace context providers
- Navigation state management integration
- Permission system integration
- Responsive design system integration
- Mobile gesture handling integration

Deliverables

- Enhanced TopNavigation with workspace context
- SideNavigation with workspace navigation
- MobileNav with workspace features
- Responsive navigation system
- Accessibility compliance validation
- Comprehensive Storybook stories

Component Specifications

typescript

```
interface TopNavigationProps {  
  context?: 'consultant' | 'client' | 'admin' | 'neutral'  
  workspaces?: Workspace[]  
  currentWorkspace?: Workspace  
  onWorkspaceChange?: (workspace: Workspace) => void  
  user?: User  
  notifications?: Notification[]  
  onNotificationClick?: (notification: Notification) => void  
  primaryActions?: NavigationAction[]  
  secondaryActions?: NavigationAction[]  
  responsive?: boolean  
}
```

```
interface SideNavigationProps {  
  context?: 'consultant' | 'client' | 'admin' | 'neutral'  
  items: NavigationItem[]  
  currentPath?: string  
  onItemClick?: (item: NavigationItem) => void  
  collapsible?: boolean  
  defaultCollapsed?: boolean  
  onCollapseChange?: (collapsed: boolean) => void  
  permissions?: string[]  
  responsive?: boolean  
}
```

```
interface MobileNavProps {  
  context?: 'consultant' | 'client' | 'admin' | 'neutral'  
  items: NavigationItem[]  
  currentPath?: string  
  onItemClick?: (item: NavigationItem) => void  
  workspaces?: Workspace[]  
  currentWorkspace?: Workspace  
  onWorkspaceChange?: (workspace: Workspace) => void  
  user?: User  
  bottomSheet?: boolean  
  swipeGestures?: boolean  
}
```

```
interface NavigationItem {  
  id: string  
  label: string  
  icon?: string  
  path?: string
```

```
children?: NavigationItem[]
permission?: string
badge?: number
active?: boolean
disabled?: boolean
workspaceContext?: 'consultant' | 'client' | 'admin' | 'neutral'
}
```

```
interface NavigationAction {
  id: string
  label: string
  icon?: string
  onClick?: () => void
  permission?: string
  badge?: number
  disabled?: boolean
}
```

Story 4.1.2: Secondary Navigation

Overview

Create contextual secondary navigation organisms including breadcrumbs and tab navigation that provide detailed navigation within specific workspace areas.

AI Developer Prompt

You are enhancing secondary navigation organisms for THE WHEEL design system. Building on the primary navigation from Story 4.1.1, you need to create contextual navigation components that provide detailed navigation within specific workspace areas.

Context

- Enhanced primary navigation with workspace context
- Need contextual navigation for specific workspace areas
- Must support dynamic breadcrumb generation
- Tab navigation needs workspace-specific features
- Secondary navigation provides detailed context

Requirements

1. Enhance BreadcrumbNav with workspace paths:

- Dynamic breadcrumb generation from workspace routes
- Workspace-specific path formatting
- Click-to-navigate functionality
- Responsive breadcrumb behavior
- Accessibility improvements

2. Enhance TabNavigation with workspace tabs:

- Workspace-specific tab sets
- Tab state management
- Dynamic tab loading
- Tab overflow handling
- Responsive tab behavior

3. Create context-aware navigation behavior:

- Navigation state persistence
- Workspace-specific navigation context
- Permission-based navigation visibility
- Navigation analytics integration
- Performance optimization

Specific Tasks

- ☐ Enhance BreadcrumbNav with workspace paths
- ☐ Add dynamic breadcrumb generation
- ☐ Enhance TabNavigation with workspace tabs
- ☐ Add tab state management
- ☐ Implement navigation state persistence
- ☐ Add permission-based navigation
- ☐ Create responsive navigation behavior

Documentation Required

- Secondary navigation system architecture
- Breadcrumb generation patterns
- Tab navigation implementation
- Navigation state management

- Permission-based navigation
- Responsive navigation guidelines

Testing Requirements

- Breadcrumb navigation tests
- Tab navigation functionality tests
- Navigation state persistence tests
- Permission-based navigation tests
- Responsive navigation tests
- Accessibility compliance tests
- Cross-browser compatibility tests

Integration Points

- Integration with primary navigation system
- Navigation state management integration
- Permission system integration
- Routing system integration
- Analytics integration

Deliverables

- Enhanced BreadcrumbNav with workspace paths
- TabNavigation with workspace tabs
- Navigation state persistence
- Permission-based navigation
- Responsive navigation system
- Comprehensive Storybook stories

Component Specifications

typescript

```
interface BreadcrumbNavProps {  
  context?: 'consultant' | 'client' | 'admin' | 'neutral'  
  items: BreadcrumbItem[]  
  maxItems?: number  
  separator?: React.ReactNode  
  onItemClick?: (item: BreadcrumbItem) => void  
  responsive?: boolean  
  generateFromPath?: boolean  
  workspaceScoped?: boolean  
}
```

```
interface TabNavigationProps {  
  context?: 'consultant' | 'client' | 'admin' | 'neutral'  
  tabs: TabItem[]  
  activeTab?: string  
  onTabChange?: (tab: TabItem) => void  
  variant?: 'line' | 'card' | 'pill'  
  size?: 'sm' | 'md' | 'lg'  
  scrollable?: boolean  
  responsive?: boolean  
  lazy?: boolean  
  permissions?: string[]  
}
```

```
interface BreadcrumbItem {  
  id: string  
  label: string  
  path?: string  
  icon?: string  
  disabled?: boolean  
  workspaceContext?: 'consultant' | 'client' | 'admin' | 'neutral'  
}
```

```
interface TabItem {  
  id: string  
  label: string  
  icon?: string  
  badge?: number  
  disabled?: boolean  
  lazy?: boolean  
  permission?: string  
  workspaceContext?: 'consultant' | 'client' | 'admin' | 'neutral'
```

```
content?: React.ReactNode  
}
```

Story 4.1.3: Workspace-Specific Navigation

Overview

Build specialized navigation organisms designed for different workspace contexts, including consultant, client, and admin-specific navigation patterns.

AI Developer Prompt

You are building workspace-specific navigation organisms for THE WHEEL design system. Building on the secondary navigation from Story 4.1.2, you need to create navigation components that are specifically designed for different workspace contexts.

Context

- Enhanced secondary navigation with workspace context
- Need navigation components for specific workspace types
- Must support different user contexts and permissions
- Navigation should adapt to workspace-specific workflows
- Each workspace type has unique navigation requirements

Requirements

1. Build WorkspaceNav component:

- Workspace-specific menu items
- Context-aware navigation hierarchy
- Workspace feature toggles
- Permission-based menu filtering
- Workspace state integration

2. Build ClientNav component:

- Client-specific navigation items
- Limited permission navigation
- Project-focused navigation
- Communication-focused layout

- Simplified navigation structure

3. Build ConsultantNav component:

- Consultant-specific navigation items
- Advanced feature access
- Client management navigation
- Billing and time tracking integration
- Analytics and reporting navigation

Specific Tasks

- ☐ Build WorkspaceNav component
- ☐ Add workspace-specific menu items
- ☐ Build ClientNav component
- ☐ Add client-specific navigation
- ☐ Build ConsultantNav component
- ☐ Add consultant-specific navigation
- ☐ Implement context switching logic
- ☐ Add permission-based navigation

Documentation Required

- Workspace-specific navigation architecture
- Context switching implementation
- Permission-based navigation patterns
- Workspace navigation customization
- Navigation accessibility guidelines
- Workspace-specific user flows

Testing Requirements

- Workspace navigation functionality tests
- Context switching tests
- Permission-based navigation tests
- Workspace-specific feature tests
- Accessibility compliance tests
- User flow tests

- Cross-browser compatibility tests

Integration Points

- Integration with workspace context providers
- Permission system integration
- Workspace state management integration
- Feature toggle system integration
- Analytics integration

Deliverables

- WorkspaceNav component with context awareness
- ClientNav component with client-specific features
- ConsultantNav component with consultant features
- Context switching logic
- Permission-based navigation
- Comprehensive Storybook stories

Component Specifications

typescript

```
interface WorkspaceNavProps {  
  workspace: Workspace  
  context: 'consultant' | 'client' | 'admin' | 'neutral'  
  user: User  
  permissions: string[]  
  features: FeatureToggle[]  
  onNavigate?: (path: string) => void  
  onWorkspaceChange?: (workspace: Workspace) => void  
  responsive?: boolean  
  collapsed?: boolean  
}
```

```
interface ClientNavProps {  
  client: Client  
  projects: Project[]  
  currentProject?: Project  
  onProjectChange?: (project: Project) => void  
  onNavigate?: (path: string) => void  
  permissions: string[]  
  responsive?: boolean  
  notifications?: Notification[]  
}
```

```
interface ConsultantNavProps {  
  consultant: Consultant  
  clients: Client[]  
  activeWorkspaces: Workspace[]  
  onClientSelect?: (client: Client) => void  
  onWorkspaceSelect?: (workspace: Workspace) => void  
  onNavigate?: (path: string) => void  
  permissions: string[]  
  responsive?: boolean  
  analytics?: AnalyticsData  
}
```

```
interface Workspace {  
  id: string  
  name: string  
  type: 'consultant' | 'client' | 'admin'  
  permissions: string[]  
  features: FeatureToggle[]  
  branding?: WorkspaceBranding  
  navigation: NavigationItem[]  
}
```

```
}

interface FeatureToggle {
  id: string
  name: string
  enabled: boolean
  permission?: string
  workspaceContext?: 'consultant' | 'client' | 'admin' | 'neutral'
}
```

Timeline and Dependencies

Timeline

- Week 1: Story 4.1.1 - Primary Navigation
- Week 2: Story 4.1.2 - Secondary Navigation
- Week 3: Story 4.1.3 - Workspace-Specific Navigation

Dependencies

- Epic 3.3 (Interactive Molecules) - Complete
- Epic 2.4 (Icon & Asset System) - Complete
- Workspace context system operational
- Permission system established
- Routing system configured

Success Metrics

- All navigation components support workspace contexts
- 100% keyboard navigation support
- Mobile navigation performs at 60fps
- Accessibility compliance (WCAG 2.1 AA)
- Navigation state persistence working
- Complete test coverage (90%+ for all components)

Risk Mitigation

- Progressive enhancement for complex navigation
- Performance monitoring for mobile navigation

- Regular accessibility audits
- Cross-device testing for touch interactions
- User testing for navigation patterns
- Clear documentation for navigation architecture