# Systems UI Components & Interactions

Detailed component specifications and interaction patterns for the Systems management interface, designed for modern web applications with enterprise security requirements.

## 🧩 Core Component Library

### 1. System Status Cards

// Quick Stats Dashboard Cards  
<StatsCard>  
 <StatsCard.Icon variant="systems" />  
 <StatsCard.Value>1,247</StatsCard.Value>  
 <StatsCard.Label>Total Systems</StatsCard.Label>  
 <StatsCard.Trend direction="up" value="+12" />  
 <StatsCard.Subtitle>vs last month</StatsCard.Subtitle>  
</StatsCard>  
  
// Variants: primary, success, warning, danger, info  
// Sizes: small, medium, large  
// States: loading, error, default

### 2. Advanced Search Component

<SearchBar>  
 <SearchBar.Input   
 placeholder="Search systems by name, IP, or description..."  
 onSearch={handleSearch}  
 debounce={300}  
 />  
 <SearchBar.Filters>  
 <FilterDropdown   
 label="Status"   
 options={statusOptions}  
 multiple={true}  
 />  
 <FilterDropdown   
 label="Risk Level"   
 options={riskLevels}  
 multiple={true}  
 />  
 <FilterDropdown   
 label="System Type"   
 options={systemTypes}  
 multiple={true}  
 />  
 <DateRangePicker   
 label="Last Scan"  
 presets={['today', 'week', 'month']}  
 />  
 </SearchBar.Filters>  
 <SearchBar.Actions>  
 <Button variant="ghost" onClick={clearFilters}>Clear All</Button>  
 <Button variant="primary" onClick={saveFilter}>Save Filter</Button>  
 </SearchBar.Actions>  
</SearchBar>

### 3. Enterprise Data Grid

<DataGrid  
 data={systemsData}  
 loading={isLoading}  
 selectable={true}  
 sortable={true}  
 filterable={true}  
 exportable={true}  
 pagination="virtual"  
 rowHeight={56}  
 onRowClick={handleRowClick}  
 onSelectionChange={handleSelection}  
>  
 <DataGrid.Column   
 field="name"   
 header="System Name"  
 sortable={true}  
 filterable={true}  
 width={200}  
 minWidth={150}  
 render={({ value, row }) => (  
 <SystemNameCell   
 name={value}  
 status={row.status}  
 classification={row.classification}  
 />  
 )}  
 />  
 <DataGrid.Column   
 field="type"   
 header="Type"  
 sortable={true}  
 filterable={true}  
 width={120}  
 render={({ value }) => (  
 <Badge variant="outline">{value}</Badge>  
 )}  
 />  
 <DataGrid.Column   
 field="status"   
 header="Status"  
 sortable={true}  
 filterable={true}  
 width={100}  
 render={({ value }) => (  
 <StatusIndicator status={value} />  
 )}  
 />  
 <DataGrid.Column   
 field="riskLevel"   
 header="Risk"  
 sortable={true}  
 filterable={true}  
 width={80}  
 render={({ value }) => (  
 <RiskBadge level={value} />  
 )}  
 />  
 <DataGrid.Column   
 field="lastScan"   
 header="Last Scan"  
 sortable={true}  
 width={120}  
 render={({ value }) => (  
 <TimeAgo date={value} />  
 )}  
 />  
 <DataGrid.Column   
 field="actions"   
 header=""  
 width={60}  
 render={({ row }) => (  
 <RowActions   
 actions={getRowActions(row)}  
 onAction={handleRowAction}  
 />  
 )}  
 />  
</DataGrid>

### 4. System Details Modal

<Modal   
 size="large"   
 isOpen={isDetailsOpen}  
 onClose={closeDetails}  
 title={selectedSystem?.name}  
 subtitle={`${selectedSystem?.type} • ${selectedSystem?.environment}`}  
>  
 <Modal.Header>  
 <SystemStatusBadge status={selectedSystem?.status} />  
 <RiskBadge level={selectedSystem?.riskLevel} />  
 <ClassificationBanner level={selectedSystem?.classification} />  
 </Modal.Header>  
   
 <Modal.Body>  
 <TabNavigation defaultTab="overview">  
 <Tab id="overview" label="Overview" icon="📊">  
 <SystemOverview system={selectedSystem} />  
 </Tab>  
 <Tab id="security" label="Security" icon="🔒">  
 <SecurityPosture system={selectedSystem} />  
 </Tab>  
 <Tab id="compliance" label="Compliance" icon="📋">  
 <ComplianceStatus system={selectedSystem} />  
 </Tab>  
 <Tab id="vulnerabilities" label="Vulnerabilities" icon="🔍">  
 <VulnerabilityList systemId={selectedSystem?.id} />  
 </Tab>  
 <Tab id="analytics" label="Analytics" icon="📈">  
 <SystemAnalytics system={selectedSystem} />  
 </Tab>  
 <Tab id="audit" label="Audit Log" icon="📝">  
 <AuditLog resourceId={selectedSystem?.id} />  
 </Tab>  
 </TabNavigation>  
 </Modal.Body>  
   
 <Modal.Footer>  
 <Button variant="ghost" onClick={closeDetails}>Close</Button>  
 <Button variant="outline" onClick={exportSystem}>Export</Button>  
 <Button variant="primary" onClick={editSystem}>Edit System</Button>  
 </Modal.Footer>  
</Modal>

### 5. Create/Edit System Slide Panel

<SlidePanel   
 isOpen={isFormOpen}  
 onClose={closeForm}  
 title={isEditing ? "Edit System" : "Create New System"}  
 size="medium"  
 position="right"  
>  
 <SlidePanel.Header>  
 <ProgressIndicator   
 steps={['Basic Info', 'Technical', 'Security', 'Review']}  
 currentStep={currentStep}  
 />  
 </SlidePanel.Header>  
   
 <SlidePanel.Body>  
 <Form onSubmit={handleSubmit} validation={systemSchema}>  
 <StepContainer currentStep={currentStep}>  
   
 <Step id="basic">  
 <FormField label="System Name" required>  
 <TextInput   
 name="name"  
 placeholder="Enter system name"  
 validation="required|min:3|max:100"  
 />  
 </FormField>  
   
 <FormField label="Description">  
 <TextArea   
 name="description"  
 placeholder="Describe the system purpose and function"  
 rows={3}  
 />  
 </FormField>  
   
 <FormField label="System Type" required>  
 <Select   
 name="type"  
 options={systemTypeOptions}  
 placeholder="Select system type"  
 />  
 </FormField>  
   
 <FormField label="Environment" required>  
 <RadioGroup   
 name="environment"  
 options={[  
 { value: 'production', label: 'Production' },  
 { value: 'staging', label: 'Staging' },  
 { value: 'development', label: 'Development' }  
 ]}  
 />  
 </FormField>  
 </Step>  
   
 <Step id="technical">  
 <FormField label="Primary IP Address">  
 <TextInput   
 name="ipAddress"  
 placeholder="192.168.1.100"  
 validation="ip"  
 />  
 </FormField>  
   
 <FormField label="Hostname/FQDN">  
 <TextInput   
 name="hostname"  
 placeholder="server.domain.com"  
 validation="fqdn"  
 />  
 </FormField>  
   
 <FormField label="Operating System">  
 <Select   
 name="operatingSystem"  
 options={osOptions}  
 searchable={true}  
 />  
 </FormField>  
   
 <FormField label="Network Location">  
 <Select   
 name="networkLocation"  
 options={networkOptions}  
 />  
 </FormField>  
 </Step>  
   
 <Step id="security">  
 <FormField label="Security Classification" required>  
 <Select   
 name="classification"  
 options={classificationOptions}  
 />  
 </FormField>  
   
 <FormField label="Impact Level (FIPS 199)" required>  
 <ImpactLevelSelector name="impactLevel" />  
 </FormField>  
   
 <FormField label="ATO Status">  
 <Select   
 name="atoStatus"  
 options={atoStatusOptions}  
 />  
 </FormField>  
   
 <FormField label="Compliance Requirements">  
 <MultiSelect   
 name="complianceRequirements"  
 options={complianceOptions}  
 />  
 </FormField>  
 </Step>  
   
 <Step id="review">  
 <ReviewSummary formData={formData} />  
 </Step>  
   
 </StepContainer>  
 </Form>  
 </SlidePanel.Body>  
   
 <SlidePanel.Footer>  
 <Button   
 variant="ghost"   
 onClick={previousStep}  
 disabled={currentStep === 0}  
 >  
 Previous  
 </Button>  
 <Button variant="ghost" onClick={closeForm}>Cancel</Button>  
 <Button variant="outline" onClick={saveDraft}>Save Draft</Button>  
 {currentStep < 3 ? (  
 <Button variant="primary" onClick={nextStep}>Next</Button>  
 ) : (  
 <Button variant="primary" onClick={submitForm}>  
 {isEditing ? 'Update System' : 'Create System'}  
 </Button>  
 )}  
 </SlidePanel.Footer>  
</SlidePanel>

## 🎨 Specialized Components

### System Status Indicator

<StatusIndicator status="active">  
 <StatusIndicator.Dot color="green" pulse={true} />  
 <StatusIndicator.Label>Active</StatusIndicator.Label>  
 <StatusIndicator.Timestamp>Last seen 2m ago</StatusIndicator.Timestamp>  
</StatusIndicator>  
  
// Status variants: active, inactive, warning, error, maintenance, unknown

### Risk Level Badge

<RiskBadge level="high">  
 <RiskBadge.Icon name="alert-triangle" />  
 <RiskBadge.Label>High Risk</RiskBadge.Label>  
 <RiskBadge.Score>8.5</RiskBadge.Score>  
</RiskBadge>  
  
// Risk levels: low, medium, high, critical  
// Colors: green, yellow, orange, red

### Classification Banner

<ClassificationBanner level="confidential">  
 <ClassificationBanner.Text>  
 CONFIDENTIAL - AUTHORIZED PERSONNEL ONLY  
 </ClassificationBanner.Text>  
</ClassificationBanner>  
  
// Levels: unclassified, confidential, secret, top-secret  
// Positions: top, bottom, both

### Bulk Action Bar

<BulkActionBar   
 selectedCount={selectedSystems.length}  
 isVisible={selectedSystems.length > 0}  
>  
 <BulkActionBar.Counter>  
 {selectedSystems.length} systems selected  
 </BulkActionBar.Counter>  
   
 <BulkActionBar.Actions>  
 <Button variant="outline" onClick={bulkAddTags}>  
 <Icon name="tag" /> Add Tags  
 </Button>  
 <Button variant="outline" onClick={bulkUpdateStatus}>  
 <Icon name="status" /> Update Status  
 </Button>  
 <Button variant="outline" onClick={bulkRunScan}>  
 <Icon name="scan" /> Run Scan  
 </Button>  
 <Button variant="outline" onClick={bulkExport}>  
 <Icon name="download" /> Export  
 </Button>  
 <Button variant="danger" onClick={bulkDelete}>  
 <Icon name="trash" /> Delete  
 </Button>  
 </BulkActionBar.Actions>  
   
 <BulkActionBar.Clear onClick={clearSelection}>  
 <Icon name="x" /> Clear Selection  
 </BulkActionBar.Clear>  
</BulkActionBar>

## 🔄 Interaction Patterns

### Loading States

// Grid Loading  
<DataGrid loading={true}>  
 <DataGrid.Skeleton rows={10} />  
</DataGrid>  
  
// Card Loading  
<StatsCard loading={true}>  
 <StatsCard.Skeleton />  
</StatsCard>  
  
// Form Loading  
<Form loading={isSubmitting}>  
 <Form.Overlay>  
 <Spinner size="large" />  
 <Text>Creating system...</Text>  
 </Form.Overlay>  
</Form>

### Error States

// Grid Error  
<DataGrid error={error}>  
 <DataGrid.ErrorState   
 title="Failed to load systems"  
 message={error.message}  
 action={<Button onClick={retry}>Retry</Button>}  
 />  
</DataGrid>  
  
// Form Validation Errors  
<FormField error={fieldError}>  
 <TextInput name="name" />  
 <FormField.Error>{fieldError}</FormField.Error>  
</FormField>

### Success Feedback

// Toast Notifications  
toast.success("System created successfully", {  
 action: {  
 label: "View System",  
 onClick: () => viewSystem(newSystem.id)  
 }  
});  
  
// Inline Success  
<Alert variant="success" dismissible={true}>  
 <Alert.Icon name="check-circle" />  
 <Alert.Title>System Updated</Alert.Title>  
 <Alert.Description>  
 Changes to {systemName} have been saved successfully.  
 </Alert.Description>  
</Alert>

## 📱 Responsive Behavior

### Mobile Layout

// Mobile System Card  
<SystemCard mobile={true}>  
 <SystemCard.Header>  
 <SystemCard.Name>{system.name}</SystemCard.Name>  
 <SystemCard.Status status={system.status} />  
 </SystemCard.Header>  
   
 <SystemCard.Body>  
 <SystemCard.Meta>  
 <MetaItem label="Type" value={system.type} />  
 <MetaItem label="Risk" value={<RiskBadge level={system.riskLevel} />} />  
 <MetaItem label="Last Scan" value={<TimeAgo date={system.lastScan} />} />  
 </SystemCard.Meta>  
 </SystemCard.Body>  
   
 <SystemCard.Actions>  
 <SwipeActions>  
 <SwipeAction color="blue" onClick={() => viewSystem(system.id)}>  
 <Icon name="eye" />  
 </SwipeAction>  
 <SwipeAction color="green" onClick={() => editSystem(system.id)}>  
 <Icon name="edit" />  
 </SwipeAction>  
 <SwipeAction color="red" onClick={() => deleteSystem(system.id)}>  
 <Icon name="trash" />  
 </SwipeAction>  
 </SwipeActions>  
 </SystemCard.Actions>  
</SystemCard>

### Tablet Layout

// Hybrid grid-card layout for tablets  
<ResponsiveGrid breakpoint="tablet">  
 <ResponsiveGrid.Desktop>  
 <DataGrid {...gridProps} />  
 </ResponsiveGrid.Desktop>  
   
 <ResponsiveGrid.Tablet>  
 <CardGrid>  
 {systems.map(system => (  
 <SystemCard key={system.id} system={system} />  
 ))}  
 </CardGrid>  
 </ResponsiveGrid.Tablet>  
   
 <ResponsiveGrid.Mobile>  
 <SystemList>  
 {systems.map(system => (  
 <SystemListItem key={system.id} system={system} />  
 ))}  
 </SystemList>  
 </ResponsiveGrid.Mobile>  
</ResponsiveGrid>

## 🎯 Accessibility Features

### Keyboard Navigation

// Grid keyboard support  
<DataGrid  
 onKeyDown={handleKeyDown}  
 tabIndex={0}  
 role="grid"  
 aria-label="Systems data grid"  
>  
 {/\* Grid supports arrow keys, Enter, Space, Tab navigation \*/}  
</DataGrid>  
  
// Modal keyboard trapping  
<Modal  
 trapFocus={true}  
 closeOnEscape={true}  
 initialFocus="[data-autofocus]"  
>  
 {/\* Focus is trapped within modal \*/}  
</Modal>

### Screen Reader Support

// Semantic markup and ARIA labels  
<main role="main" aria-label="Systems Management">  
 <section aria-labelledby="stats-heading">  
 <h2 id="stats-heading" className="sr-only">System Statistics</h2>  
 <StatsCard aria-label="Total systems: 1,247" />  
 </section>  
   
 <section aria-labelledby="systems-heading">  
 <h2 id="systems-heading" className="sr-only">Systems List</h2>  
 <DataGrid   
 aria-label="Systems data table"  
 aria-describedby="systems-description"  
 />  
 <div id="systems-description" className="sr-only">  
 Use arrow keys to navigate, Enter to view details  
 </div>  
 </section>  
</main>

This component library provides a comprehensive foundation for building a modern, accessible, and user-friendly Systems management interface that meets enterprise security requirements while maintaining excellent usability.