

CareLinkAI Persona Dashboards

Overview

Each role in CareLinkAI has a personalized dashboard that surfaces the most relevant information and quick actions for that persona. This document provides a comprehensive overview of all persona-specific dashboards.

Architecture

Role-Based Routing

When users navigate to `/dashboard`, they are automatically redirected to their role-specific dashboard:

- **FAMILY** → `/dashboard` (renders `FamilyDashboard` component)
- **OPERATOR** → `/operator`
- **CAREGIVER** → `/caregiver`
- **PROVIDER** → `/provider`
- **ADMIN** → `/admin`

This routing logic is implemented in `src/app/dashboard/page.tsx`.

Reusable Components

Two shared components provide consistent UI across all dashboards:

StatTile Component

Location: `src/components/dashboard/StatTile.tsx`

Displays a metric tile with:

- Icon (emoji or React icon)
- Numeric or text value
- Title and description
- Optional link for navigation
- Optional trend indicator

Props:

```
{
  title: string;
  value: number | string;
  icon: ReactNode;
  description: string;
  href?: string;
  trend?: { value: string; isPositive: boolean };
}
```

QuickActionCard Component

Location: `src/components/dashboard/QuickActionCard.tsx`

Displays a clickable action card with:

- Large icon (emoji or React icon)
- Title and description
- Link to destination

Props:

```
{
  title: string;
  description: string;
  href: string;
  icon: ReactNode;
}
```

Dashboards by Role

Family Dashboard

Route: /dashboard (for FAMILY role)

Component: `src/app/dashboard/FamilyDashboard.tsx`

Purpose: Help families find care and track their inquiries.

Tiles:

1. **Home Inquiries** - Count of active care home requests

- Query: `Inquiry.count({ where: { familyId, status not in [PLACEMENT_ACCEPTED, CLOSED_LOST] } })`

- Links to: `/dashboard/inquiries`

1. **Aide/Provider Requests** - Count of open caregiver inquiries

- Query: `Lead.count({ where: { familyId, status not in [CLOSED, CANCELLED] } })`

2. **Total Activity** - Sum of all active requests

- Calculation: `homeInquiries + aideProviderRequests`

Quick Actions:

- **Search Homes** → `/search-homes`

- Find care homes in your area

- **Find Caregivers** → `/marketplace?tab=caregivers`

- Browse available caregivers

- **Find Providers** → `/marketplace?tab=providers`

- Explore service providers

Recent Activity:

- List of 5 most recent inquiries (home + aide/provider combined)

- Shows:

- Type icon (🏠 home, 👤 aide, 🏢 provider)

- Target name

- Status badge

- Created date

- Link to detail page

- Empty state with CTA to search homes if no activity

Data Sources:

- Inquiry model - home inquiries
 - Lead model - aide/provider inquiries
 - Combined and sorted by creation date
-

Operator Dashboard**Route:** /operator**Component:** src/app/operator/page.tsx**Purpose:** Help operators manage leads, caregivers, residents, and homes.**Tiles:**1. **Homes** - Count of all homes managed

- Query: `AssistedLivingHome.count({ where: { operatorId } })`
- Links to: /operator/homes

1. **Open Inquiries** - Count of all inquiries

- Query: `Inquiry.count({ where: { home.operatorId } })`
- Shows count of NEW status inquiries
- Links to: /operator/inquiries

2. **Active Residents** - Count of current residents

- Query: `Resident.count({ where: { home.operatorId, status: ACTIVE } })`
- Links to: /operator/residents

3. **Occupancy Rate** - Percentage of filled beds

- Calculation: $(\text{sum currentOccupancy} / \text{sum capacity}) * 100$
- Color-coded: red (<50%), yellow (50-80%), green (>80%)

Critical Alerts:

- Displays banner for:
- New inquiries awaiting response
- Licenses expiring within 30 days
- Links to relevant pages for action

Recent Activity:

- Table of 5 most recent inquiries
- Shows:
- Family name
- Home name
- Status badge
- Created date
- Links to detail page

Quick Actions:

- **Add Home** → /operator/homes/new
- Create new listing
- **Add Resident** → /operator/residents/new
- Onboard new resident
- **View Inquiries** → /operator/inquiries
- Manage leads

Additional Links:

- Manage Homes → `/operator/homes`
- Analytics → `/operator/analytics`
- Compliance → `/operator/compliance`
- Billing → `/operator/billing`

Special Features:

- Admin users can filter by operatorId via query parameter
- Mock mode support via cookie

Caregiver Dashboard

Route: `/caregiver`**Component:** `src/app/caregiver/page.tsx`**Purpose:** Help caregivers manage their profile and track opportunities.**Tiles:****1. Profile Visibility** - Shows if visible in marketplace

- Query: `Caregiver.isVisibleInMarketplace`
- Value: "Visible" or "Hidden"
- Links to: `/settings/profile`

1. Background Check - Verification status

- Query: `Caregiver.backgroundCheckStatus`
- Values: VERIFIED, PENDING, REJECTED, NOT_STARTED
- Color-coded badge

2. Active Requests - Count of open family inquiries

- Query: `Lead.count({ where: { aideId, status not in [CLOSED, CANCELLED] } })`

Profile Visibility Alert:

- Yellow banner shown if profile is hidden
- Encourages user to make profile visible
- Links to profile settings

Recent Activity:

- List of 5 most recent leads from families
- Shows:
 - Family name
 - Message preview
 - Status badge
 - Created date
- Empty state with CTA to complete profile

Quick Actions:

- **Edit Profile** → `/settings/profile`
- Update bio, skills, and availability
- **Upload Documents** → `/settings/credentials`
- Add certifications and credentials
- **Messages** → `/messages`
- Check conversations

Data Sources:

- `Caregiver` model - profile and visibility
 - `Lead` model (targetType: AIDE) - family inquiries
-

Provider Dashboard**Route:** `/provider`**Component:** `src/app/provider/page.tsx`**Purpose:** Help providers manage their services and track inquiries.**Tiles:****1. New Inquiries (7 days)** - Count of recent requests

- Query: `Lead.count({ where: { providerId, status: NEW, createdAt >= 7 days ago } })`

1. Active Inquiries - Count of open conversations

- Query: `Lead.count({ where: { providerId, status not in [CLOSED, CANCELLED] } })`

2. Verification Status - Business verification

- Query: `Provider.isVerified`
- Values: VERIFIED, PENDING
- Color-coded badge

Verification Alert:

- Yellow banner shown if not verified
- Encourages completion of profile and credentials
- Links to provider settings

Recent Activity:

- List of 5 most recent leads from families
- Shows:
 - Family name
 - Location
 - Message preview
 - Status badge
 - Created date
- Empty state with CTA to complete profile

Quick Actions:

- **Edit Profile** → `/settings/provider`
- Update business info and services
- **Upload Documents** → `/settings/provider/credentials`
- Add licenses and insurance
- **Messages** → `/messages`
- Check conversations

Data Sources:

- `Provider` model - business info and verification
 - `Lead` model (targetType: PROVIDER) - family inquiries
-

Admin Dashboard

Route: `/admin`

Component: `src/app/admin/page.tsx`

Purpose: Provide system overview and quick access to admin tools.

Tiles:

1. **Total Users** - Count of all registered users

- Query: `User.count()`

1. **Total Inquiries** - Count of all inquiries (home + leads)

- Query: `Inquiry.count() + Lead.count()`

2. **Active Caregivers** - Count of marketplace-visible aides

- Query: `Caregiver.count({ where: { isVisibleInMarketplace: true } })`

- Links to: `/admin/aides`

3. **Verified Providers** - Count of active service providers

- Query: `Provider.count({ where: { isVerified: true } })`

- Links to: `/admin/providers`

Pending Actions Alert:

- Amber banner shown if there are unverified credentials

- Shows count of credentials awaiting review

- Links to:

- `/admin/aides` - Review caregiver credentials

- `/admin/providers` - Review provider credentials

Platform Management:

Six quick access cards:

1. **Caregivers** → `/admin/aides`

- Manage aide profiles and credentials

2. **Providers** → `/admin/providers`

- Manage service provider verification

3. **Analytics** → `/admin/metrics`

- Detailed metrics and charts

4. **Tools** → `/admin/tools`

- Admin utilities and settings

5. **Operator View** → `/operator`

- Switch to operator dashboard

System Information:

- Display panel with:

- Total home inquiries

- Total marketplace listings (aides + providers)

- Platform operational status

Data Sources:

- `User` model - all users

- `Inquiry` and `Lead` models - all inquiries

- `Caregiver` and `Provider` models - marketplace status

- `Credential` model - pending verifications

Technical Implementation

Database Queries

All dashboards use Prisma ORM for database access:

```
import { prisma } from "@lib/prisma";
```

Queries are scoped to the logged-in user:

- Family: `familyId` from session
- Caregiver: `caregiverId` from user.id lookup
- Provider: `providerId` from user.id lookup
- Operator: `operatorId` from user.id lookup (admins can view all)
- Admin: Global queries (no scope)

Authentication & Authorization

All dashboards enforce RBAC:

```
const session = await getServerSession(authOptions);

if (!session?.user) {
  redirect('/auth/login');
}

if (session.user.role !== 'EXPECTED_ROLE') {
  redirect('/unauthorized');
}
```

Server Components

All dashboards are React Server Components:

- Data fetching happens server-side
- No client-side state needed
- Optimal performance with Next.js 14 App Router

Error Handling

Dashboards gracefully handle:

- Missing user records (return default values)
- Database query failures (return empty arrays)
- No data scenarios (show empty states with CTAs)

Styling & Design

Design System

- **Tailwind CSS** for all styling
- **Consistent spacing:** p-6 for main container, mb-8 for sections
- **Responsive grid:** 1 column mobile, 2-3 columns tablet, 3-4 columns desktop
- **Color scheme:**

- Blue (#3B82F6) for primary actions
- Green (#10B981) for success/positive
- Yellow (#F59E0B) for warnings
- Red (#EF4444) for errors/critical
- Gray scale for neutral elements

Card Styling

StatTile and QuickActionCard share consistent styling:

- White background
- Gray border (border-gray-200)
- Rounded corners (rounded-lg)
- Hover effects (border-blue-500, shadow-md)
- Transition animations

Status Badges

Color-coded badges for inquiry/lead status:

- NEW: Blue (bg-blue-100 text-blue-700)
- CONTACTED: Yellow (bg-yellow-100 text-yellow-700)
- IN_REVIEW: Orange (bg-orange-100 text-orange-700)
- Other: Gray (bg-gray-100 text-gray-700)

Icons

Emoji icons used for simplicity and universal recognition:

- 🏠 - Homes
- 👥 - Caregivers
- 🏢 - Providers
- 📋 - Inquiries/Leads
- ✅ - Success/Verified
- ⚠️ - Warnings
- 💬 - Messages
- ✎️ - Edit
- 📄 - Documents

Testing

Manual Testing Checklist

For each persona dashboard:

1. Authentication

- [] Requires login
- [] Redirects unauthenticated users
- [] Blocks wrong roles (shows unauthorized)

2. Data Display

- [] Shows correct metrics
- [] Handles no data gracefully
- [] Recent activity displays correctly
- [] Status badges show proper colors

3. Navigation

- [] Quick action links work
- [] Tile links navigate correctly
- [] Detail page links work
- [] Back navigation works

4. Responsive Design

- [] Mobile (320px-768px) layout works
- [] Tablet (768px-1024px) layout works
- [] Desktop (>1024px) layout works

5. Edge Cases

- [] New user with no activity
- [] User with hidden profile
- [] Unverified user
- [] Large numbers (>999)

Test Accounts

Use demo accounts for testing:

- **Family:** `demo.family@carelinkai.test / DemoUser123!`
 - **Operator:** `demo.operator@carelinkai.test / DemoUser123!`
 - **Admin:** `demo.admin@carelinkai.test / DemoUser123!`
 - **Caregiver:** Check aide marketplace for test accounts
 - **Provider:** Check provider marketplace for test accounts
-

Performance Considerations

Database Optimization

- All queries use indexed fields
- Count queries are efficient (no SELECT *)
- Limited list queries (TAKE 3-5)
- Selective includes (only needed relations)

Server-Side Rendering

Benefits:

- Faster initial page load
- SEO-friendly (though dashboards are private)
- No loading spinners for initial data
- Reduced client-side JavaScript

Caching Strategy

- Use Next.js App Router automatic caching
 - Mark pages as dynamic where needed (`export const dynamic = "force-dynamic"`)
 - Consider implementing ISR for less-critical dashboards
-

Future Enhancements

Phase 2 Features (Not in MVP)

1. Real-Time Updates

- WebSocket integration for live notifications
- Auto-refresh for new inquiries
- Presence indicators

2. Advanced Analytics

- Charts and graphs for trends
- Comparative metrics (week-over-week, month-over-month)
- Conversion funnels

3. Customization

- User-configurable widgets
- Drag-and-drop dashboard builder
- Saved filters and preferences

4. Enhanced Empty States

- Onboarding wizards
- Step-by-step guides
- Interactive tutorials

5. Notifications Center

- In-dashboard notification feed
- Read/unread tracking
- Bulk actions

6. Export Functionality

- Download inquiry data as CSV
- Generate PDF reports
- Email summaries

Deployment Checklist

Before deploying to production:

1. ☐ All TypeScript errors resolved
 2. ☐ Build succeeds (`npm run build`)
 3. ☐ Manual testing completed for all roles
 4. ☐ Database migrations applied
 5. ☐ Environment variables configured
 6. ☐ RBAC tested thoroughly
 7. ☐ Mobile responsiveness verified
 8. ☐ Error boundaries in place
 9. ☐ Logging configured
 10. ☐ Analytics tracking added (if applicable)
-

Troubleshooting

Common Issues

Issue: Dashboard shows empty state but user has data

- **Solution:** Check database query scope (familyId, caregiverId, etc.)
- **Verify:** User role matches expected role in query

Issue: TypeScript errors about missing properties

- **Solution:** Ensure Prisma schema is up to date (`npx prisma generate`)
- **Verify:** Model includes expected fields

Issue: Redirect loop

- **Solution:** Check middleware and page-level redirects
- **Verify:** Auth session is valid

Issue: Tile/card not clickable

- **Solution:** Verify `href` prop is passed and valid
- **Check:** Link component is rendering correctly

Issue: Status badges wrong color

- **Solution:** Check status value against enum
 - **Verify:** Color mapping function handles all cases
-

Maintenance

Regular Updates

- Review dashboard metrics relevance quarterly
- Update quick actions based on user behavior
- Refine empty states based on conversion data
- Optimize queries if performance degrades

Monitoring

Key metrics to track:

- Dashboard load time
 - Click-through rates on quick actions
 - Empty state frequency by role
 - Tile interaction patterns
-

Related Documentation

- [MVP Status Matrix](#) (./mvp_status_matrix.md)
 - [Demo Flow Guide](#) (./DEMO_FLOW.md)
 - [Family Lead Schema Design](#) (./family_leads_schema_design.md)
 - [Provider Implementation](#) (./PROVIDER_MVP_IMPLEMENTATION_SUMMARY.md)
-

Support

For questions or issues with persona dashboards:

1. Check this documentation first
 2. Review related code files
 3. Test with demo accounts
 4. Check database schema and migrations
 5. Review Prisma query logs
-

Last Updated: December 8, 2024

Version: 1.0

Status: Production Ready