

Operator Lead Management Implementation

Overview

This document describes the implementation of Phase 5: Operator Lead Management APIs & UI for the CareLinkAI Family Leads MVP. This phase enables operators to efficiently view, filter, and manage leads generated from family inquiries about aides and providers.

Implementation Date

December 7, 2025

Branch

`feature/family-leads-mvp`

Components

1. API Endpoints

1.1 GET /api/operator/leads

Purpose: Retrieve paginated list of leads with filtering capabilities

Location: `src/app/api/operator/leads/route.ts`

Features:

- RBAC enforcement (OPERATOR or ADMIN roles only)
- Multiple filter support:
 - `status` : Single or multiple LeadStatus values (NEW, IN_REVIEW, CONTACTED, CLOSED, CANCELLED)
 - `targetType` : AIDE or PROVIDER
 - `assignedOperatorId` : Filter by operator assignment (supports "unassigned", "me", or specific operator ID)
- Pagination:
 - `page` : Page number (default: 1)
 - `limit` : Items per page (default: 20, max: 100)
- Sorting:
 - `sortBy` : Field to sort by (createdAt, updatedAt, status, targetType)
 - `sortOrder` : asc or desc (default: desc)
- Includes related data:
 - Family (with User info)
 - Aide or Provider (based on targetType)
 - Assigned operator

Request Example:

```
GET /api/operator/leads?status=NEW,IN_REVIEW&targetType=AIDE&page=1&limit=20
```

Response Format:

```

{
  "success": true,
  "data": {
    "leads": [
      {
        "id": "lead_123",
        "familyId": "family_123",
        "targetType": "AIDE",
        "status": "NEW",
        "message": "Looking for experienced caregiver...",
        "preferredStartDate": "2025-01-15T00:00:00Z",
        "expectedHoursPerWeek": 20,
        "location": "Seattle, WA",
        "operatorNotes": null,
        "createdAt": "2025-12-07T10:00:00Z",
        "updatedAt": "2025-12-07T10:00:00Z",
        "family": {
          "id": "family_123",
          "primaryContactName": "Jane Doe",
          "phone": "+1-206-555-0100",
          "user": {
            "firstName": "Jane",
            "lastName": "Doe",
            "email": "jane@example.com"
          }
        },
        "aide": {
          "id": "aide_123",
          "user": {
            "firstName": "John",
            "lastName": "Smith"
          }
        },
        "assignedOperator": null
      }
    ],
    "pagination": {
      "total": 45,
      "pages": 3,
      "currentPage": 1,
      "limit": 20,
      "hasMore": true
    }
  }
}

```

1.2 GET /api/operator/leads/[id]

Purpose: Retrieve detailed information about a specific lead

Location: `src/app/api/operator/leads/[id]/route.ts`

Features:

- RBAC enforcement (OPERATOR or ADMIN roles only)
- Complete lead details with all relationships
- Audit logging for lead views
- Excludes soft-deleted leads

Request Example:

```
GET /api/operator/leads/lead_123
```

Response Format:

```
{
  "success": true,
  "data": {
    "id": "lead_123",
    "familyId": "family_123",
    "targetType": "AIDE",
    "status": "NEW",
    "message": "Looking for experienced caregiver...",
    "preferredStartDate": "2025-01-15T00:00:00Z",
    "expectedHoursPerWeek": 20,
    "location": "Seattle, WA",
    "operatorNotes": null,
    "createdAt": "2025-12-07T10:00:00Z",
    "updatedAt": "2025-12-07T10:00:00Z",
    "family": {
      "id": "family_123",
      "primaryContactName": "Jane Doe",
      "phone": "+1-206-555-0100",
      "relationshipToRecipient": "Daughter",
      "recipientAge": 78,
      "primaryDiagnosis": "Alzheimer's disease",
      "mobilityLevel": "Walker",
      "careNotes": "Patient needs assistance with ADLs...",
      "user": {
        "id": "user_123",
        "email": "jane@example.com",
        "firstName": "Jane",
        "lastName": "Doe",
        "phone": "+1-206-555-0100"
      }
    },
    "aide": {
      "id": "aide_123",
      "user": {
        "id": "user_456",
        "firstName": "John",
        "lastName": "Smith",
        "email": "john@example.com",
        "phone": "+1-206-555-0101",
        "profileImageUrl": null
      }
    },
    "provider": null,
    "assignedOperator": null
  }
}
```

1.3 PATCH /api/operator/leads/[id]

Purpose: Update lead status, operator notes, or assignment

Location: `src/app/api/operator/leads/[id]/route.ts`

Features:

- RBAC enforcement (OPERATOR or ADMIN roles only)

- Zod validation for input data
- Validates operator assignment (must be OPERATOR or ADMIN role)
- Audit logging for changes
- Tracks changed fields in audit log

Request Body Schema:

```
{
  status?: LeadStatus,           // NEW, IN_REVIEW, CONTACTED, CLOSED, CANCELLED
  operatorNotes?: string,        // Max 5000 characters
  assignedOperatorId?: string    // Must be valid OPERATOR or ADMIN user ID
}
```

Request Example:

```
PATCH /api/operator/leads/lead_123
Content-Type: application/json

{
  "status": "IN_REVIEW",
  "operatorNotes": "Called family, scheduling follow-up...",
  "assignedOperatorId": "operator_123"
}
```

Response Format:

```
{
  "success": true,
  "message": "Lead updated successfully",
  "data": {
    // Updated lead object with all relations
  }
}
```

Error Responses:

- 400: Validation error or invalid operator assignment
- 401: Not authenticated
- 403: Not authorized (not OPERATOR or ADMIN)
- 404: Lead not found

2. UI Components

2.1 LeadStatusBadge

Purpose: Display lead status with color-coded badges

Location: `src/components/operator/LeadStatusBadge.tsx`

Props:

- `status` : LeadStatus (required)
- `size` : "sm" | "md" | "lg" (optional, default: "md")
- `className` : string (optional)

Color Scheme:

- NEW: Blue (bg-blue-100, text-blue-800)

- IN_REVIEW: Yellow (bg-yellow-100, text-yellow-800)
- CONTACTED: Green (bg-green-100, text-green-800)
- CLOSED: Gray (bg-gray-100, text-gray-800)
- CANCELLED: Red (bg-red-100, text-red-800)

Usage Example:

```
<LeadStatusBadge status="NEW" size="sm" />
```

2.2 LeadTargetTypeBadge

Purpose: Display lead target type (AIDE or PROVIDER) with icons

Location: src/components/operator/LeadTargetTypeBadge.tsx

Props:

- targetType : LeadTargetType (required)
- size : "sm" | "md" | "lg" (optional, default: "md")
- showIcon : boolean (optional, default: true)
- className : string (optional)

Color Scheme:

- AIDE: Purple (bg-purple-100, text-purple-800)
- PROVIDER: Teal (bg-teal-100, text-teal-800)

Icons:

- AIDE: FiUser
- PROVIDER: FiUsers

Usage Example:

```
<LeadTargetTypeBadge targetType="AIDE" showIcon={true} />
```

2.3 LeadFilters

Purpose: Reusable filter component for lead lists

Location: src/components/operator/LeadFilters.tsx

Props:

- filters : LeadFiltersState (required)
- onFilterChange : (filters: LeadFiltersState) => void (required)
- operatorId : string (optional, for "Assigned to me" filter)
- className : string (optional)

Filter Types:

1. **Search:** Text input for family name or lead ID
2. **Status:** Multi-select tabs (All, New, In Review, Contacted, Closed, Cancelled)
3. **Target Type:** Dropdown (All Types, Aides, Providers)
4. **Assignment:** Dropdown (All, Unassigned, Assigned to Me)

Usage Example:

```
<LeadFilters
  filters={filters}
  onFilterChange={handleFilterChange}
  operatorId={session?.user?.id}
/>
```

3. UI Pages

3.1 Operator Leads List Page

Purpose: Display paginated list of leads with filtering

Location: `src/app/operator/leads/page.tsx`

Features:

- RBAC: OPERATOR or ADMIN only (enforced by API)
- Responsive design:
 - Desktop: Table view with all columns
 - Mobile: Card view with essential info
- Filter panel using LeadFilters component
- Real-time filtering and pagination
- Client-side search (family name, contact name, lead ID)
- Empty states with helpful messages
- Loading indicators
- Error handling with retry option

Columns (Desktop Table):

- Lead ID (first 8 characters)
- Family (contact name)
- Target (aide/provider name)
- Type (badge)
- Status (badge)
- Created (date with icon)
- Assigned To (operator name or “Unassigned”)
- Actions (View button)

Mobile Card Fields:

- Family name and target name
- Type and status badges
- Created date
- Assigned operator
- View Details button

Pagination:

- Page indicator (e.g., “Page 1 of 3”)
- Previous/Next buttons
- Disabled states for first/last pages

3.2 Operator Lead Detail Page

Purpose: Display comprehensive lead information with editing capabilities

Location: `src/app/operator/leads/[id]/page.tsx`

Features:

- RBAC: OPERATOR or ADMIN only (enforced by API)
- Comprehensive information display
- Inline editing of status, notes, and assignment
- Success/error message display
- Link back to leads list
- Profile links to aide/provider

Layout:

- **Header:** Back button, badges (type and status)
- **Summary Card:** Lead ID, created date, last updated, assignment
- **Left Column** (2/3 width):
 - Family Information (contact, email, phone, relationship)
 - Target Information (name, contact, profile link)
 - Inquiry Details (start date, hours/week, location, message)
 - Care Context (age, diagnosis, mobility, care notes)
- **Right Column** (1/3 width):
 - Status Update dropdown
 - Assignment dropdown
 - Operator Notes textarea (5000 char limit)
 - Save Changes button
 - Open Conversation button (placeholder for Phase 6)

Form Validation:

- Character limit for operator notes (5000)
- Tracks changes before saving
- Displays success/error messages
- Disables save button while saving

User Experience:

- Auto-populates fields with current values
- Only sends changed fields to API
- Displays character count for notes
- Shows loading states during save
- Links to marketplace profiles for aides/providers

4. Navigation Integration

Updated File: `src/components/layout/DashboardLayout.tsx`

Changes:

- Added "Leads" navigation item
- Icon: FiUsers
- Route: `/operator/leads`
- Role restriction: `["OPERATOR", "ADMIN"]`
- Position: After "Inquiries" in navigation list

Navigation Item:

```
{
  name: "Leads",
  icon: <FiUsers size={20} />,
  href: "/operator/leads",
  showInMobileBar: false,
  roleRestriction: ["OPERATOR", "ADMIN"]
}
```

Database Schema

Lead Model

Already implemented in Phase 1. Key fields:

- `id` : Unique identifier
- `familyId` : Reference to Family
- `targetType` : AIDE or PROVIDER
- `aideId` : Reference to Caregiver (when `targetType=AIDE`)
- `providerId` : Reference to Provider (when `targetType=PROVIDER`)
- `status` : NEW, IN_REVIEW, CONTACTED, CLOSED, CANCELLED
- `message` : Inquiry message from family
- `preferredStartDate` : When family wants service to start
- `expectedHoursPerWeek` : Expected hours of service
- `location` : Service location
- `operatorNotes` : Internal notes for operators
- `assignedOperatorId` : Reference to assigned operator User
- `deletedAt` : Soft delete timestamp

Relationships

- `family` : Family record with User details
- `aide` : Caregiver record (when `targetType=AIDE`)
- `provider` : Provider record (when `targetType=PROVIDER`)
- `assignedOperator` : User record (OPERATOR or ADMIN)

RBAC Implementation

Roles Allowed

- OPERATOR
- ADMIN

Enforcement

- API level: `requireAnyRole(["OPERATOR", "ADMIN"])`
- UI level: Navigation item with `roleRestriction: ["OPERATOR", "ADMIN"]`
- Session validation on all API calls

Permissions

- View all leads (list and detail)
- Update lead status
- Add/edit operator notes
- Assign leads to operators

- View family and target information

Audit Logging

Events Logged

1. **Lead View** (READ):
 - Action: AuditAction.READ
 - Resource: "LEAD"
 - Description: "Operator viewed lead details"
 - Includes: resourceId, userId, ipAddress
2. **Lead Update** (UPDATE):
 - Action: AuditAction.UPDATE
 - Resource: "LEAD"
 - Description: Lists all changed fields
 - Metadata: Changed fields, previous/new status
 - Includes: resourceId, userId, ipAddress

Audit Log Fields

- `action` : READ or UPDATE
- `resourceType` : "LEAD"
- `resourceId` : Lead ID
- `description` : Human-readable description
- `metadata` : Additional context (changed fields, values)
- `userId` : Operator who performed action
- `actionedBy` : Same as userId
- `ipAddress` : Client IP address
- `createdAt` : Timestamp

Testing

API Testing

1. GET /api/operator/leads

```
# Test basic list
curl -X GET "http://localhost:3000/api/operator/leads" \
  -H "Cookie: session_token"

# Test with filters
curl -X GET "http://localhost:3000/api/operator/leads?
status=NEW&targetType=AIDE&page=1&limit=10" \
  -H "Cookie: session_token"

# Test assignment filter
curl -X GET "http://localhost:3000/api/operator/leads?assignedOperatorId=unassigned" \
  -H "Cookie: session_token"

# Test "assigned to me" filter
curl -X GET "http://localhost:3000/api/operator/leads?assignedOperatorId=me" \
  -H "Cookie: session_token"
```

2. GET /api/operator/leads/[id]

```
# Test detail view
curl -X GET "http://localhost:3000/api/operator/leads/lead_123" \
  -H "Cookie: session_token"

# Test non-existent lead (should return 404)
curl -X GET "http://localhost:3000/api/operator/leads/invalid_id" \
  -H "Cookie: session_token"
```

3. PATCH /api/operator/leads/[id]

```
# Test status update
curl -X PATCH "http://localhost:3000/api/operator/leads/lead_123" \
  -H "Content-Type: application/json" \
  -H "Cookie: session_token" \
  -d '{"status": "IN_REVIEW"}'

# Test notes update
curl -X PATCH "http://localhost:3000/api/operator/leads/lead_123" \
  -H "Content-Type: application/json" \
  -H "Cookie: session_token" \
  -d '{"operatorNotes": "Called family, scheduling follow-up..."}'

# Test assignment
curl -X PATCH "http://localhost:3000/api/operator/leads/lead_123" \
  -H "Content-Type: application/json" \
  -H "Cookie: session_token" \
  -d '{"assignedOperatorId": "operator_123"}'

# Test multiple updates
curl -X PATCH "http://localhost:3000/api/operator/leads/lead_123" \
  -H "Content-Type: application/json" \
  -H "Cookie: session_token" \
  -d '{
    "status": "CONTACTED",
    "operatorNotes": "Scheduled meeting for next week",
    "assignedOperatorId": "operator_123"
  }'
```

4. RBAC Testing

```
# Test as non-operator (should return 403)
curl -X GET "http://localhost:3000/api/operator/leads" \
  -H "Cookie: family_session_token"

# Test without authentication (should return 401)
curl -X GET "http://localhost:3000/api/operator/leads"
```

UI Testing

Leads List Page

1. Navigation:

- [] Verify “Leads” link appears in navigation for OPERATOR/ADMIN
- [] Verify “Leads” link does NOT appear for FAMILY/CAREGIVER
- [] Click “Leads” link navigates to /operator/leads

2. List Display:

- [] Verify leads list loads on page mount
- [] Verify table displays on desktop (lg breakpoint)
- [] Verify cards display on mobile
- [] Verify all columns/fields are visible
- [] Verify status badges display with correct colors
- [] Verify target type badges display correctly

3. Filtering:

- [] Verify search input filters by family name
- [] Verify status filter tabs work (All, New, In Review, etc.)
- [] Verify target type dropdown filters correctly
- [] Verify assignment dropdown filters correctly
- [] Verify "Assigned to Me" filter works
- [] Verify multiple filters work together
- [] Verify "Clear Filters" button resets all filters

4. Pagination:

- [] Verify page indicator shows current page
- [] Verify Previous button is disabled on first page
- [] Verify Next button is disabled on last page
- [] Verify clicking Next/Previous changes page
- [] Verify pagination controls only show when needed

5. Actions:

- [] Verify "View" button navigates to detail page
- [] Verify clicking row on mobile navigates to detail page

6. States:

- [] Verify loading spinner displays while fetching
- [] Verify empty state displays with no results
- [] Verify empty state message changes based on filters
- [] Verify error state displays on API error
- [] Verify "Try Again" button refetches data

Lead Detail Page**1. Navigation:**

- [] Verify "Back to Leads" button navigates to list
- [] Verify type and status badges display in header

2. Information Display:

- [] Verify summary card shows all fields
- [] Verify family information section displays correctly
- [] Verify target information section displays correctly
- [] Verify inquiry details section displays correctly
- [] Verify care context section displays (when available)
- [] Verify care context section is hidden (when not available)

3. Form Functionality:

- [] Verify status dropdown populates with current status
- [] Verify assignment dropdown populates with current assignment
- [] Verify operator notes textarea populates with current notes

- [] Verify character count updates as user types
- [] Verify character limit enforced (5000 chars)

4. **Update Operations:**

- [] Verify “Save Changes” button is enabled
- [] Verify clicking “Save Changes” updates lead
- [] Verify success message displays on successful update
- [] Verify error message displays on failed update
- [] Verify save button shows loading state while saving
- [] Verify save button is disabled while saving
- [] Verify no API call when no changes made

5. **Links:**

- [] Verify email links open mail client
- [] Verify profile links open in new tab
- [] Verify profile links go to correct marketplace page

6. **States:**

- [] Verify loading spinner displays while fetching
- [] Verify error state displays on API error
- [] Verify 404 state for non-existent lead

7. **Responsive:**

- [] Verify layout is responsive on mobile
- [] Verify all sections stack vertically on mobile
- [] Verify form controls work on mobile

Integration Testing

Complete Operator Workflow

1. Log in as OPERATOR
2. Navigate to “Leads” from navigation menu
3. Verify leads list displays
4. Apply filters (e.g., status=NEW, targetType=AIDE)
5. Verify filtered results
6. Click “View” on a lead
7. Verify detail page displays all information
8. Update status to “IN_REVIEW”
9. Add operator notes
10. Assign lead to self
11. Click “Save Changes”
12. Verify success message
13. Navigate back to list
14. Verify lead shows updated status and assignment
15. Filter by “Assigned to Me”
16. Verify lead appears in filtered list

Cross-Phase Integration

1. **Phase 4 → Phase 5:** Family creates lead → Operator views lead

2. **Phase 5 → Phase 6:** Operator views lead → Opens conversation (future)

Error Handling

API Errors

- 400: Invalid input (validation errors with field details)
- 401: Not authenticated (redirect to login)
- 403: Not authorized (show access denied message)
- 404: Lead not found (show error state)
- 500: Server error (show error message with retry option)

UI Error States

- Network errors: Display error message with “Try Again” button
- Validation errors: Display field-specific error messages
- Empty results: Display helpful empty state
- Loading failures: Display error with context

Validation

- Status: Must be valid LeadStatus enum value
- Operator notes: Max 5000 characters
- Assigned operator: Must be valid OPERATOR or ADMIN user
- Lead ID: Must exist and not be soft-deleted

Security Considerations

RBAC

- All API endpoints enforce OPERATOR or ADMIN role
- Navigation item only visible to authorized roles
- Session validation on every request

Data Access

- Operators can view all leads (no scoping to specific operators)
- Audit logs track all view and update operations
- Soft delete prevents accidental data loss

Input Validation

- Zod schemas for all input validation
- Character limits on text fields
- Enum validation for status and targetType
- User role validation for assignments

HIPAA Compliance

- Care context data is protected
- Audit logs track all access
- No PHI in URLs or query parameters
- Secure transmission (HTTPS in production)

Future Enhancements

Phase 6: Messaging Integration

- “Open Conversation” button functionality
- Direct messaging between operator and family
- Automated notifications for lead updates

Additional Features (Post-MVP)

- **Lead Analytics Dashboard:**

- Lead volume trends
- Conversion rates by status
- Average response times
- Operator performance metrics

- **Lead Assignment Rules:**

- Auto-assignment based on criteria
- Round-robin distribution
- Workload balancing

- **Advanced Filtering:**

- Date range filters (created, updated)
- Care context filters (age, diagnosis, mobility)
- Location-based filtering
- Hours/week range filter

- **Bulk Actions:**

- Select multiple leads
- Bulk status updates
- Bulk assignment
- Export to CSV

- **Activity Feed:**

- Chronological history of changes
- Operator actions timeline
- Status change history
- Notes history

- **Notifications:**

- New lead alerts
- Assignment notifications
- Status change notifications
- Follow-up reminders

- **Lead Scoring:**

- Priority scoring algorithm
- Urgency indicators
- Match quality score
- **Reports:**
 - Lead source analysis
 - Operator productivity reports
 - Conversion funnel analysis
 - Time-to-contact metrics

Deployment Checklist

Pre-Deployment

- ☐ All API endpoints tested locally
- ☐ All UI pages tested locally
- ☐ RBAC tested for all roles
- ☐ Audit logging verified
- ☐ Error handling tested
- ☐ Responsive design verified
- ☐ Browser compatibility checked

Deployment

- ☐ Code committed to feature branch
- ☐ Pull request created
- ☐ Code review completed
- ☐ Merge to main branch
- ☐ Deploy to staging environment
- ☐ Run smoke tests on staging
- ☐ Deploy to production

Post-Deployment

- ☐ Verify API endpoints accessible
- ☐ Verify navigation link appears for operators
- ☐ Test complete workflow end-to-end
- ☐ Monitor error logs
- ☐ Check audit logs
- ☐ Verify performance metrics

Maintenance

Monitoring

- Monitor API response times
- Track error rates
- Review audit logs regularly
- Monitor database query performance

Updates

- Keep dependencies updated
- Review and optimize database indexes
- Refactor based on usage patterns
- Address user feedback

Documentation Updates

Files Updated

1. `operator_lead_management_implementation.md` (this file)
2. `family_leads_schema_design.md` (reference)
3. `PHASE1_IMPLEMENTATION_SUMMARY.md` (reference)

Related Documentation

- Phase 1: Database schema (Family and Lead models)
- Phase 2: Family profile APIs and UI
- Phase 3: Family registration with onboarding
- Phase 4: Lead creation and inquiry submission
- Phase 6: Messaging integration (upcoming)

Support

Common Issues

Issue: Leads not displaying

- Check RBAC (must be OPERATOR or ADMIN)
- Verify API endpoint is accessible
- Check browser console for errors
- Verify database connection

Issue: Filters not working

- Check query parameter format
- Verify API endpoint accepts filter parameters
- Check browser console for errors
- Clear filters and try again

Issue: Save not working








- Check validation errors
- Verify operator assignment is valid
- Check character limits
- Verify network connectivity

Issue: Navigation link not visible

- Check user role (must be OPERATOR or ADMIN)
- Verify session is valid
- Clear browser cache
- Re-login if necessary

Conclusion

Phase 5 successfully implements a comprehensive operator lead management system with:

-  Robust API endpoints with filtering and pagination
-  Intuitive UI with responsive design
-  Comprehensive RBAC enforcement
-  Audit logging for compliance
-  Efficient database queries with proper indexing
-  Extensive error handling
-  Complete documentation

The system provides operators with the tools they need to efficiently manage family inquiries, track lead status, and coordinate care services. The implementation follows best practices for security, performance, and user experience.

Next phase will integrate messaging capabilities to enable direct communication between operators and families.