

Phase 6: Caregiver Management System - Deployment Ready

Deployment Summary

The Caregiver Management System (Phase 6: Parts 1 & 2) is ready for production deployment.

Implementation Status

Phase 6 Part 1 (Complete)

- ☒ Database schema updates
- ☒ New models: Caregiver, CaregiverCertification, CaregiverAssignment, CaregiverDocument
- ☒ Migration: `20251209220507_add_caregiver_management`
- ☒ RBAC permissions for caregivers
- ☒ 14 API endpoints implemented
- ☒ Committed: `b1bdbcb`

Phase 6 Part 2 (Complete)

- ☒ 12 UI components
- ☒ Caregiver list and detail pages
- ☒ 4 tabbed interfaces
- ☒ 3 modal forms
- ☒ Navigation updates
- ☒ Compliance dashboard
- ☒ RBAC integration
- ☒ Responsive design

Deployment Strategy

Auto-Deployment via Render

When code is pushed to GitHub, Render will automatically:

1. Pull latest code from `main` branch
2. Install dependencies
3. Run database migrations
4. Generate Prisma client
5. Build Next.js application
6. Deploy to production

Expected Deployment Time: 5-8 minutes

Manual Deployment (If Needed)

If auto-deployment doesn't trigger:

1. Go to Render dashboard: <https://dashboard.render.com>

2. Navigate to CareLinkAI service
3. Click “Manual Deploy” > “Deploy latest commit”

Database Changes

Migration Applied (Part 1)

- **Migration:** 20251209220507_add_caregiver_management
- **Applied:** Yes (Part 1 deployment)
- **Status:** Idempotent, safe to re-run

Changes Made:

1. **Caregiver Table**
 - Basic profile fields
 - Employment information
 - Specializations (array)
 - Languages (array)
 - Experience tracking
2. **CaregiverCertification Table**
 - Certification tracking
 - Expiration dates
 - Issuing organization
 - Document links
3. **CaregiverAssignment Table**
 - Resident assignments
 - Primary/backup designation
 - Start/end dates
 - Assignment notes
4. **CaregiverDocument Table**
 - Document management
 - Expiration tracking
 - Document URLs

No New Migrations (Part 2)

Part 2 only added UI components - no database changes required.

API Endpoints (14 Total)

Caregiver Management (4 endpoints)

- GET /api/operator/caregivers - List caregivers
- POST /api/operator/caregivers - Create caregiver
- GET /api/operator/caregivers/[id] - Get caregiver details
- PATCH /api/operator/caregivers/[id] - Update caregiver
- DELETE /api/operator/caregivers/[id] - Delete caregiver

Certification Management (4 endpoints)

- GET /api/operator/caregivers/[id]/certifications - List certifications

- `POST /api/operator/caregivers/[id]/certifications` - Create certification
- `PATCH /api/operator/caregivers/[id]/certifications/[certId]` - Update certification
- `DELETE /api/operator/caregivers/[id]/certifications/[certId]` - Delete certification

Assignment Management (3 endpoints)

- `GET /api/operator/caregivers/[id]/assignments` - List assignments
- `POST /api/operator/caregivers/[id]/assignments` - Create assignment
- `DELETE /api/operator/caregivers/[id]/assignments/[assignmentId]` - Delete assignment

Document Management (4 endpoints)

- `GET /api/operator/caregivers/[id]/documents` - List documents
- `POST /api/operator/caregivers/[id]/documents` - Create document
- `PATCH /api/operator/caregivers/[id]/documents/[docId]` - Update document
- `DELETE /api/operator/caregivers/[id]/documents/[docId]` - Delete document

Compliance Dashboard (1 endpoint)

- `GET /api/operator/caregivers/compliance` - Compliance overview

RBAC Permissions

New Permissions Added:

```
// Caregivers
CAREGIVERS_VIEW: "caregivers.view",
CAREGIVERS_CREATE: "caregivers.create",
CAREGIVERS_UPDATE: "caregivers.update",
CAREGIVERS_DELETE: "caregivers.delete",

// Certifications
CERTIFICATIONS_VIEW: "certifications.view",
CERTIFICATIONS_CREATE: "certifications.create",
CERTIFICATIONS_UPDATE: "certifications.update",
CERTIFICATIONS_DELETE: "certifications.delete",

// Assignments
ASSIGNMENTS_VIEW: "assignments.view",
ASSIGNMENTS_CREATE: "assignments.create",
ASSIGNMENTS_DELETE: "assignments.delete",

// Documents
DOCUMENTS_VIEW: "documents.view",
DOCUMENTS_CREATE: "documents.create",
DOCUMENTS_UPDATE: "documents.update",
DOCUMENTS_DELETE: "documents.delete",
```

Role Assignments:

- **ADMIN:** All permissions
- **OPERATOR:** All caregiver-related permissions
- **CAREGIVER:** View own profile only
- **FAMILY:** No caregiver management access

Post-Deployment Verification

1. Database Verification

Run these SQL queries to verify schema:

```
-- Check Caregiver table exists
SELECT COUNT(*) FROM "Caregiver";

-- Check Certification table exists
SELECT COUNT(*) FROM "CaregiverCertification";

-- Check Assignment table exists
SELECT COUNT(*) FROM "CaregiverAssignment";

-- Check Document table exists
SELECT COUNT(*) FROM "CaregiverDocument";
```

2. API Endpoint Testing

Test with curl or Postman:

```
# Test caregiver list (requires auth token)
curl -X GET https://carelinkai.onrender.com/api/operator/caregivers \
  -H "Cookie: next-auth.session-token=YOUR_TOKEN"

# Test compliance dashboard
curl -X GET https://carelinkai.onrender.com/api/operator/caregivers/compliance \
  -H "Cookie: next-auth.session-token=YOUR_TOKEN"
```

3. UI Verification

1. Login as Operator/Admin

- URL: <https://carelinkai.onrender.com>
- Navigate to Operator → Caregivers

2. Test Caregiver List Page

- Page loads without errors
- Search functionality works
- Filters work correctly
- “Add Caregiver” button visible

3. Test Create Caregiver

- Click “Add Caregiver”
- Fill out form
- Submit successfully
- Verify caregiver appears in list

4. Test Caregiver Detail Page

- Click on a caregiver
- All 4 tabs load correctly
- Overview shows all information
- Edit button works

5. Test Certifications

- Navigate to Certifications tab
- Click "Add Certification"
- Fill out form
- Submit successfully
- Verify status badge correct

6. Test Assignments

- Navigate to Assignments tab
- Click "Assign Resident"
- Select resident
- Submit successfully
- Verify assignment appears

7. Test Documents

- Navigate to Documents tab
- Click "Add Document"
- Fill out form
- Submit successfully
- Verify document appears

4. RBAC Verification

Test with different user roles:

- **As ADMIN:** Should have full access
- **As OPERATOR:** Should have full access to own caregivers
- **As CAREGIVER:** Should see own profile only (if implemented)
- **As FAMILY:** Should not see Caregivers menu item

5. Mobile Responsiveness

- Test on mobile device or DevTools
- List page should show single column
- Forms should be scrollable
- Tabs should be scrollable horizontally
- All buttons should be easily tappable

Rollback Plan

If Critical Issues Occur:

1. Revert Code

```
bash
git revert HEAD
git push origin main
```

2. Rollback Migration (If Needed)

```
```sql
-- Run this SQL in production database
BEGIN;
```

```
DROP TABLE IF EXISTS "CaregiverDocument" CASCADE;
DROP TABLE IF EXISTS "CaregiverAssignment" CASCADE;
DROP TABLE IF EXISTS "CaregiverCertification" CASCADE;
DROP TABLE IF EXISTS "Caregiver" CASCADE;
```

```
COMMIT;
```

```

1. **Restore Previous State**

- Render will auto-deploy previous version
- Wait for deployment to complete
- Verify system is stable

Monitoring

Key Metrics to Watch:

1. **API Response Times**

- Caregiver list endpoint
- Detail page endpoints
- Create/Update operations

2. **Error Rates**

- Check Render logs for 500 errors
- Monitor database errors
- Watch for permission errors

3. **User Activity**

- Track how many caregivers are created
- Monitor certification additions
- Check assignment frequency

Render Dashboard:

- URL: <https://dashboard.render.com>
- Check "Logs" tab for errors
- Monitor "Metrics" for performance
- Review "Events" for deployment status

Known Limitations

1. **Demo Data:** Seed script needs type adjustments
2. **Photo Upload:** Currently URL-based, not file upload
3. **Document Upload:** Currently URL-based, not file upload
4. **Compliance API:** Needs implementation

Support Contacts

For Technical Issues:

- Check Render logs
- Review GitHub commits









- Consult implementation docs

For User Issues:







- Verify RBAC permissions
- Check data scoping
- Review user roles

Success Criteria

Deployment is successful if:

-  Build completes without errors
-  All API endpoints respond with 200/201
-  UI pages load without errors
-  Forms submit successfully
-  RBAC permissions enforced
-  No critical errors in logs
-  Mobile layout works
-  Data persists correctly

Deployment requires attention if:

-  500 errors on API calls
-  Permission errors for valid users
-  UI components not rendering
-  Forms failing to submit
-  Database connection errors
-  Migration failures

Next Steps After Deployment

1. Monitor for 24 hours

- Watch Render logs
- Check error rates
- Monitor user feedback

2. Create Demo Data

- Use UI to create test caregivers
- Or fix seed script and run

3. User Training

- Document new features
- Train operators on caregiver management
- Provide user guides

4. Future Enhancements

- Implement file uploads
- Add email notifications
- Create mobile app support
- Add bulk operations

Phase 6 Status:  READY FOR DEPLOYMENT

Deployment Method: Auto-deploy on GitHub push

Estimated Deploy Time: 5-8 minutes

Risk Level: Low (No breaking changes)