

Caregiver Detail Resident Relation Fix

Issue

Error: PrismaClientValidationError: Unknown field 'user' for include statement on model 'Resident'

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

Root Cause: The API was attempting to include a `user` relation on the `Resident` model in the caregiver assignments section, but the Resident model doesn't have a `user` relation.

Root Cause Analysis

Schema Reality

According to `prisma/schema.prisma`, the `Resident` model has direct fields for name information:

```
model Resident {  
  id          String      @id @default(cuid())  
  familyId    String  
  homeId      String?  
  firstName   String      // ✓ Direct field  
  lastName    String      // ✓ Direct field  
  dateOfBirth DateTime  
  gender      String  
  status      ResidentStatus @default(INQUIRY)  
  // ... other fields  
}
```

Key Point: Unlike the `Caregiver` model which has a `user` relation, the `Resident` model stores name information directly.

API Error

The caregiver detail API was incorrectly trying to access resident names through a non-existent `user` relation:

```
// ❌ WRONG - This structure doesn't exist
assignments: {
  where: { endDate: null },
  include: {
    resident: {
      include: {
        user: { // ❌ ERROR: No such relation
          select: {
            firstName: true,
            lastName: true,
          }
        }
      }
    }
  }
}
```

Solution

1. Fixed Prisma Query Structure

File: /src/app/api/operator/caregivers/[id]/route.ts

Before:

```
assignments: {
  where: {
    endDate: null // Only active assignments
  },
  include: {
    resident: {
      include: {
        user: {
          select: {
            firstName: true,
            lastName: true,
          }
        }
      }
    }
  }
}
```

After:

```

assignments: {
  where: {
    endDate: null // Only active assignments
  },
  include: {
    resident: {
      select: {
        id: true,
        firstName: true,      // ✓ Direct field
        lastName: true,      // ✓ Direct field
        status: true,
        photoUrl: true,
      }
    }
  }
}

```

2. Fixed Data Transformation

Before:

```

assignments: caregiver.assignments.map(assignment => ({
  ...assignment,
  resident: {
    ...assignment.resident,
    fullName: `${assignment.resident.user.firstName} ${assignment.resident.user.lastName}`
    // ✗ ERROR: No 'user' property
  }
})),

```

After:

```

assignments: caregiver.assignments.map(assignment => ({
  ...assignment,
  resident: {
    ...assignment.resident,
    fullName: `${assignment.resident.firstName} ${assignment.resident.lastName}`
    // ✓ Access fields directly
  }
})),

```

Changes Summary

Modified Files

1. `src/app/api/operator/caregivers/[id]/route.ts`
 - Fixed Prisma query to select resident fields directly
 - Updated data transformation to use correct structure

Key Changes

- ✓ Removed invalid `user` relation from Resident include
- ✓ Selected `firstName`, `lastName` directly from Resident model
- ✓ Updated response mapping to access fields directly
- ✓ Added additional useful fields (`status`, `photoUrl`)

Deployment





Git Commit

```
commit e856024
Author: Your Name
Date: Thu Dec 11 2025

fix: Correct Resident model structure in caregiver assignments query

- Remove invalid 'user' relation from Resident include
- Access firstName/lastName directly from Resident model
- Update data transformation to use correct structure
- Fixes PrismaClientValidationError on caregiver detail page
```

Deployment Status

-  Changes committed
-  Pushed to GitHub (main branch)
-  Render auto-deploy triggered
-  Expected: <https://carelinkai.onrender.com>

Verification Steps

Once deployed, verify:

1. API Endpoint Works:

```
bash
```

```
curl https://carelinkai.onrender.com/api/operator/caregivers/[caregiverId]
```

Should return 200 OK with caregiver data including assignments

2. Detail Page Loads:

- Navigate to: [https://carelinkai.onrender.com/operator/caregivers/\[caregiverId\]](https://carelinkai.onrender.com/operator/caregivers/[caregiverId])
- Page should load without errors
- Assignments tab should display resident names correctly

3. Console Check:

- No Prisma validation errors
- Resident names appear in assignments list

Expected Result

- ✓ **Prisma Query Success:** Query executes without validation errors
- ✓ **API Response Valid:** Returns caregiver data with correct assignment structure
- ✓ **Detail Page Renders:** Page loads successfully with resident information
- ✓ **Resident Names Display:** Names appear correctly in assignments table

Related Models

Caregiver → User Relation

```
model Caregiver {
  userId String @unique
  user User @relation(fields: [userId], references: [id])
  // Has user relation
}
```

Resident → No User Relation

```
model Resident {
  firstName String // Direct field
  lastName String // Direct field
  familyId String
  family Family @relation(fields: [familyId], references: [id])
  // No user relation
}
```

Lessons Learned

1. **Always verify schema structure** before writing queries
2. **Different models have different architectures:**
 - Caregiver uses User relation for personal data
 - Resident stores personal data directly
3. **Schema documentation is critical** for avoiding such errors
4. **Type-safe queries** help catch these issues early

Prevention

To prevent similar issues:

1. ☒ Always check `prisma/schema.prisma` before writing queries
2. ☒ Use Prisma's type system to catch relation errors
3. ☒ Test API endpoints thoroughly during development
4. ☒ Document model structures in API comments

Status: ☒ Fixed and deployed

Priority: High (blocking caregiver detail page)

Impact: Caregiver detail page now fully functional