

# Timesheet System Fixes - Root Cause Analysis

**Date:** December 10, 2025

**Branch:** expenses-structure

**Issues:** Timesheet Calculation, File Upload, sendToAgency Mutation

## Issue 1: Timesheet Amount Calculation

### Root Cause

The Timesheet model already has the correct fields for separate amounts:

- `baseAmount` - work amount ( $\text{hours} \times \text{rate}$ )
- `marginAmount` - calculated margin
- `totalExpenses` - sum of expenses
- `totalAmount` - final total

### The Problem:

1. The naming is confusing - there's no explicit `workAmount` field, `baseAmount` serves this purpose
2. The calculation in `createRange` mutation (line 366) is CORRECT:

`typescript`

```
finalTotalAmount = totalWithMargin.add(totalExpenses)
// where totalWithMargin = baseAmount + marginAmount
```

3. However, the **display logic** in UI components might be re-adding these values, causing duplicate totals
4. The `totalAmount` field stores the complete sum: `baseAmount + marginAmount + totalExpenses`

### Current Schema (Correct)

```
model Timesheet {
    baseAmount Decimal? @db.Decimal(10, 2)      // Work amount
    marginAmount Decimal? @db.Decimal(10, 2)        // Margin
    totalExpenses Decimal? @db.Decimal(10, 2)       // Expenses
    totalAmount Decimal? @db.Decimal(10, 2)          // Final total
}
```

### Solution

- ✅ Schema is correct - NO migration needed
- ✅ Backend calculation is correct
- ⚠️ Need to add explicit `workAmount` field for clarity (optional)
- ⚠️ Fix UI display components to show amounts without re-adding
- ⚠️ Ensure proper documentation of what each field represents

## Issue 2: File Upload During Timesheet Creation

---

### Root Cause

There are **TWO different file storage systems** for timesheets:

#### System 1: Legacy Fields (Used by creation form)

- Fields: `timesheetFileUrl`, `expenseFileUrl`
- Set during `createRange` mutation (lines 259-260)
- Uses fake upload function in form (line 42-45)
- Files NOT stored in `TimesheetDocument` table

#### System 2: TimesheetDocument Table (Used by detail page)

- Table: `TimesheetDocument`
- Created by `uploadExpenseDocument` mutation (lines 736-783)
- Used by `TimesheetDocumentUploader` component
- Properly integrated with detail page

### The Disconnect:

1. Creation form uploads to fake URLs: `https://fake-url.com/filename`
2. Files are stored in `timesheetFileUrl` and `expenseFileUrl` fields
3. Detail page queries `TimesheetDocument` table which is empty
4. Files uploaded during creation NEVER appear in the document list

### Solution

Two options:

#### Option A: Use TimesheetDocument for creation (RECOMMENDED)

- Update creation form to use real S3 upload
- Create `TimesheetDocument` records during timesheet creation
- Remove legacy `timesheetFileUrl` and `expenseFileUrl` fields
- Unified file system across creation and detail pages

#### Option B: Support both systems

- Keep legacy fields for backward compatibility
- Display both legacy files AND `TimesheetDocument` records
- Gradually migrate to `TimesheetDocument` only

---

## Issue 3: sendToAgency Mutation - MarginType Enum Error

---

### Root Cause

The error message: "Invalid value for argument `marginType`. Expected `MarginType`."

### Investigation:

1. Prisma schema has correct enum: `FIXED`, `VARIABLE`, `CUSTOM` (lines 984-988)
2. MarginService returns correct enum type (line 98)
3. `sendToAgency` uses: `marginType: marginCalculation.marginType` (line 710)

### Potential Causes:

1. **Database contains lowercase values:** The Contract table might have "fixed" instead of "FIXED"
2. **Type casting issue:** TypeScript allows string assignment to enum type during development
3. **Migration issue:** Old data might have lowercase strings before enum was enforced

### Evidence:

- In `createRange` mutation (line 254): `const marginType = contract.marginType?.toLowerCase() || "percentage"`
- This suggests the code was written to handle lowercase values
- But Prisma enums are UPPERCASE by definition

## Solution

1. **Fix enum handling** in `sendToAgency` mutation:
  - Add enum value normalization before creating Margin
  - Convert any lowercase values to uppercase
  - Add validation to ensure enum compliance
2. **Fix data inconsistency:**
  - Query database for lowercase `marginType` values
  - Create migration to convert to uppercase
  - Add validation at contract creation
3. **Update MarginService:**
  - Add enum normalization helper
  - Ensure all margin type handling uses correct enum values

## Implementation Plan

### Phase 1: Fix `sendToAgency` Enum Issue (CRITICAL)

- [ ] Add enum normalization helper
- [ ] Update `sendToAgency` to ensure uppercase enum values
- [ ] Add validation before Margin creation

### Phase 2: Fix File Upload System

- [ ] Update creation form to use real S3 upload
- [ ] Create `TimesheetDocument` records during creation
- [ ] Remove fake upload function
- [ ] Test file visibility in detail page

### Phase 3: Clarify Amount Calculation

- [ ] Add clear comments in schema about each amount field
- [ ] Update UI components to display amounts correctly
- [ ] Add helper function to calculate amounts consistently
- [ ] Document calculation logic

### Phase 4: Testing

- [ ] Test timesheet creation with files

- [ ] Test sendToAgency with invoice creation
  - [ ] Verify margin calculation and enum handling
  - [ ] Verify file visibility in detail page
  - [ ] Test amount display in UI
- 

## Expected Outcomes

1.  sendToAgency creates invoices without enum errors
2.  Files uploaded during creation appear in detail page
3.  Amount display is clear and correct (no duplicate additions)
4.  All calculations use consistent logic
5.  Margin type enum values are correct throughout system