

Timesheet & Invoice Improvements

Overview

This document describes the comprehensive improvements made to the timesheet and invoice system, implementing 5 major features requested by the user.

Implemented Features

1. Send to Agency - Automatic Invoice Creation

Location: `server/api/routers/timesheet.ts` - `sendToAgency` mutation

What it does:

- When "Send to Agency" is clicked, the system automatically creates a professional invoice for the agency linked to the contract
- The invoice appears in the agency's invoice list with all necessary payment information
- Invoice includes:
 - All timesheet entries (hours worked)
 - Expense documents with proper references
 - Margin calculations
 - Professional formatting with summary

Key Changes:

- Enhanced `sendToAgency` mutation to include expense documents
- Proper calculation of base amount ($\text{hours} \times \text{rate}$) + expenses
- Margin calculation applied correctly
- Professional invoice notes with detailed breakdown
- Line items include both work entries and expense references

2. Visual Participant Diagram in Timesheet

Location: `components/timesheets/TimesheetParticipantDiagram.tsx`

What it does:

- Displays a visual diagram showing the workflow participants
- Shows contractor (who sends) and agency (who receives) with an arrow between them
- Does NOT show detailed timesheet entries - only the participant flow
- Clean, professional design with avatars and role badges

Features:

- Contractor avatar with blue theme
- Agency avatar with purple theme
- Animated arrow showing "Sends Invoice" flow
- Company/user names and emails displayed
- Helpful description text

3. Document Management System for Timesheets

Location:

- `components/timesheets/TimesheetDocumentManager.tsx` (upload component)

- prisma/schema.prisma (new TimesheetDocument model)
- server/api/routers/timesheet.ts (new mutations)

What it does:

- Replaces the single “fakeurl” with a proper multi-document system
- Users can upload multiple expense documents (up to 10)
- Documents are displayed as “Expense File 1”, “Expense File 2”, “Expense File 3”, etc.
- Same system as contracts for consistency

Features:

- Upload multiple files (PDF, images, Word docs)
- Each file gets a sequential name: “Expense File N”
- View, download, and delete documents
- File size validation (10 MB max per file)
- S3 storage integration
- Documents linked to timesheet via foreign key

Database Schema:

```
model TimesheetDocument {
  id          String    @id @default(cuid())
  timesheetId String
  fileName    String
  fileUrl     String
  fileSize    Int
  mimeType   String?
  description String?
  category    String    @default("expense")
  uploadedAt  DateTime  @default(now())
  timesheet   Timesheet @relation(...)
}
```

API Endpoints:

- uploadExpenseDocument - Upload a new expense document
- deleteExpenseDocument - Remove an expense document
- updateTotalExpenses - Update the total expenses amount

4. Corrected totalAmount Calculation

Location: server/api/routers/timesheet.ts - sendToAgency mutation

What it does:

- Fixes the calculation to properly include expenses
- Formula: `totalAmount = (hours × hourly rate) + total expenses`

Calculation Flow:

1. Calculate base amount from hours worked:

- Hourly: `hours × rate`
- Daily: `days × rate`
- Monthly: `(days / 20) × rate`
- Fixed: `rate`

1. Add expenses:

- `subtotal = baseAmount + totalExpenses`

2. Apply margin (if configured):

- Fixed margin: `marginAmount = margin`
- Percentage margin: `marginAmount = subtotal × (margin / 100)`
- If paid by client: `total = subtotal + marginAmount`
- If paid by contractor: `total = subtotal - marginAmount`

Database Changes:

- Added `totalExpenses` field to Timesheet model
- Properly tracked in invoice creation

5. Professional Invoice Formatting

Location: `server/api/routers/timesheet.ts` - `sendToAgency` mutation

What it does:

- Creates a professionally formatted invoice with all necessary information
- Includes detailed breakdown in the notes section
- References all expense documents

Invoice Structure:

Line Items:

1. Work entries (one per day):

- Description: "Work on YYYY-MM-DD: [description]"
- Quantity: hours worked
- Unit price: hourly/daily rate
- Amount: calculated

1. Expense entries (one per document):

- Description: "Expense File N: [description]"
- References the uploaded document

2. Expense summary:

- Description: "Total Expenses (N documents)"
- Amount: total expenses

Invoice Notes:

Generated from approved timesheet.

 Summary:

- Total Hours: XX.XX
- Hourly/Daily Rate: XXX USD
- Subtotal (Hours): XXX USD
- Expenses: XXX USD (N documents)
- Base Amount: XXX USD
- Margin (X.XX%): XXX USD
- Total Amount: XXX USD

 Agency: [Agency Name]

 Expense Documents:

1. filename1.pdf
2. filename2.pdf
3. filename3.pdf

Database Changes

New Model: TimesheetDocument

```
model TimesheetDocument {
  id          String  @id @default(cuid())
  timesheetId String
  fileName    String
  fileUrl    String
  fileSize    Int
  mimeType   String?
  description String?
  category   String  @default("expense")
  uploadedAt DateTime @default(now())
  timesheet   Timesheet @relation(...)
}
```

Updated Model: Timesheet

```
model Timesheet [
  // ... existing fields ...
  totalExpenses  Decimal? @db.Decimal(10, 2) @default(0)  // NEW
  documents      TimesheetDocument[]                      // NEW
]
```

Migration

Run the migration to apply database changes:

```
npx prisma migrate deploy
```

Migration file: `prisma/migrations/20251209111202_add_timesheet_documents/migration.sql`

New Files Created

1. Components:

- `components/timesheets/TimesheetParticipantDiagram.tsx` - Visual workflow diagram
- `components/timesheets/TimesheetDocumentManager.tsx` - Document upload/management

2. Database:

- `prisma/migrations/20251209111202_add_timesheet_documents/migration.sql` - Migration

3. Documentation:

- `TIMESHEET_INVOICE_IMPROVEMENTS.md` - This file

Modified Files

1. Backend:

- `server/api/routers/timesheet.ts` - Enhanced `sendToAgency`, added document endpoints
- `prisma/schema.prisma` - Added `TimesheetDocument` model, `totalExpenses` field

2. Frontend:

- app/(dashboard)/(modules)/timesheets/[id]/page.tsx - Added diagram, improved file display

Testing Checklist

- [] Create a timesheet with multiple expense documents
- [] Verify documents show as "Expense File 1", "Expense File 2", etc.
- [] Submit and approve timesheet
- [] Click "Send to Agency"
- [] Verify invoice is created with:
 - [] All work entries
 - [] All expense documents referenced
 - [] Correct total calculation (hours + expenses)
 - [] Margin applied correctly
 - [] Professional formatting in notes
- [] Verify invoice appears in agency's invoice list
- [] Check participant diagram displays correctly
- [] Test document upload/delete functionality

UI/UX Improvements

1. Participant Diagram:

- Clean visual representation
- Color-coded roles (blue for contractor, purple for agency)
- Animated arrow for flow indication
- Helpful description text

2. Document Display:

- Sequential naming (Expense File 1, 2, 3...)
- File size and name shown
- View and download buttons
- Clean card-based layout

3. Invoice Format:

- Professional structure
- Clear breakdown of charges
- Emoji icons for visual clarity
- All relevant information included

Future Enhancements

Potential improvements for future iterations:

1. Document Categories:

- Allow categorizing expenses (travel, meals, equipment, etc.)
- Filter/group by category in invoice

2. Expense Amounts:

- Add individual expense amounts per document
- Automatic calculation of totalExpenses from documents

3. Invoice Templates:

- Customizable invoice templates
- PDF generation with company branding

4. Approval Workflow:

- Multi-level approval for high-value invoices
- Automatic notifications to agency

5. Document Preview:

- In-app document preview (PDF viewer)
- Thumbnail generation for images

 **Support**

For questions or issues related to these improvements, please contact the development team or refer to the main project documentation.

Last Updated: December 9, 2025

Version: 1.0.0

Author: AI Development Team