

TimeTrack Pro - Technical & Functional Reference

Executive Summary

TimeTrack Pro is a professional time tracking application built on Microsoft Access, showcasing advanced database design, VBA automation, and modern development practices. This project demonstrates the capability to deliver production-ready business applications through automated development workflows using the VBA MCP Server.

Key Achievement: Complete application development (database structure, business logic, queries, and VBA modules) automated via MCP (Model Context Protocol) integration, demonstrating cutting-edge AI-assisted development capabilities.

Project Overview

Purpose

TimeTrack Pro is a time management tool designed for freelancers, consultants, and small teams to:

- Track billable hours across multiple clients and projects
- Calculate revenue automatically based on hourly rates
- Generate professional reports for invoicing and analysis
- Maintain a complete audit trail of time entries

Target Audience

- **Freelancers:** Independent consultants tracking multiple client projects
- **Small Teams:** Agencies managing client work and resource allocation
- **Consultants:** Professional services requiring detailed time records

Differentiators

- Automated Development:** Built using VBA MCP Server v0.6.0+ for database automation
- Clean Architecture:** Modular VBA design with separation of concerns
- Production-Ready:** Complete with data validation, error handling, and user-friendly interfaces
- Extensible:** Well-documented codebase ready for customization and enhancement

Technical Specifications

Technology Stack

Component	Technology	Version
Database Engine	Microsoft Access	2016+ / Office 365
Programming Language	VBA (Visual Basic for Applications)	7.1
Development Automation	VBA MCP Server	v0.6.0+
Export Formats	PDF, Excel	Native Access/VBA
Version Control	Git	2.x

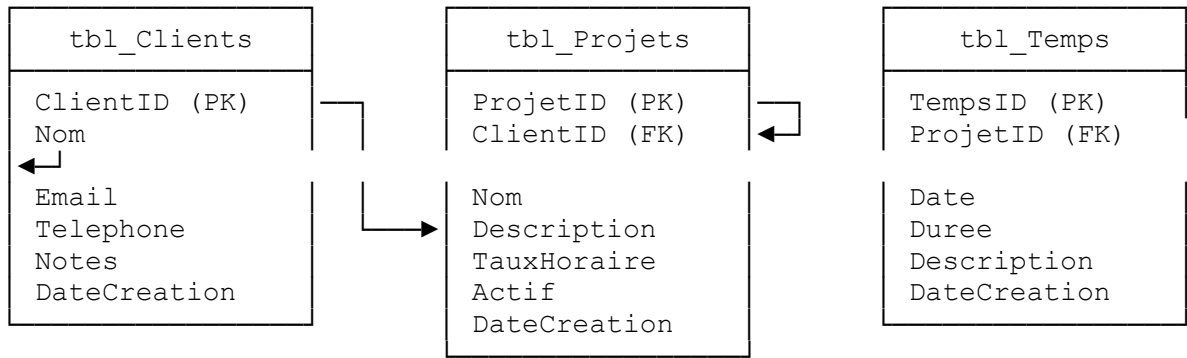
System Requirements

- **Minimum:****
- Windows 10 or later
 - Microsoft Access 2016 or Office 365 with Access
 - 100 MB disk space
 - 2 GB RAM
- **Recommended:****
- Windows 11
 - Office 365 (latest)
 - SSD storage for optimal performance

Database Architecture

Entity Relationship Diagram

...



Relationships:
tbl_Clients (1) — (N) tbl_Projets
tbl_Projets (1) — (N) tbl_Temps
...

Data Model

Table: tbl_Clients

Stores client information and contact details.

Field	Type	Size	Required	Description
ClientID	AutoNumber	Long	PK	Unique identifier
Nom	Text	100	Yes	Client name
Email	Text	100	No	Email address
Telephone	Text	20	No	Phone number
Notes	Memo	-	No	Additional notes
DateCreation	DateTime	-	Yes	Record creation timestamp

- **Indexes:****
- Primary Key: ClientID
 - Search Index: Nom

Table: tbl_Projets

Stores project information linked to clients.

Field	Type	Size	Required	Description
ProjID	AutoNumber	Long	PK	Unique identifier
ClientID	Long	-	FK	Reference to tbl_Clients
Nom	Text	100	Yes	Project name
Description	Memo	-	No	Project description
TauxHoraire	Currency	-	No	Hourly rate (EUR)
Actif	Yes/No	-	Yes	Active/archived status
DateCreation	DateTime	-	Yes	Record creation timestamp

Indexes:

- Primary Key: ProjID
- Foreign Key: ClientID (with referential integrity)
- Performance Index: Actif

Table: tbl_Temps

Stores time entry records.

Field	Type	Size	Required	Description
TempsID	AutoNumber	Long	PK	Unique identifier
ProjID	Long	-	FK	Reference to tbl_Projets
Date	DateTime	-	Yes	Entry date
Duree	Double	-	Yes	Duration in hours (decimal)
Description	Memo	-	No	Work description
DateCreation	DateTime	-	Yes	Record creation timestamp

Indexes:

- Primary Key: TempsID
- Foreign Key: ProjID (with referential integrity)
- Performance Index: Date (for date range queries)

Data Integrity

Referential Integrity Rules:

- Client deletion: Cascade to projects and time entries (configurable)
- Project deletion: Cascade to time entries (configurable)
- Orphan prevention: Foreign key constraints enforced

Validation Rules:

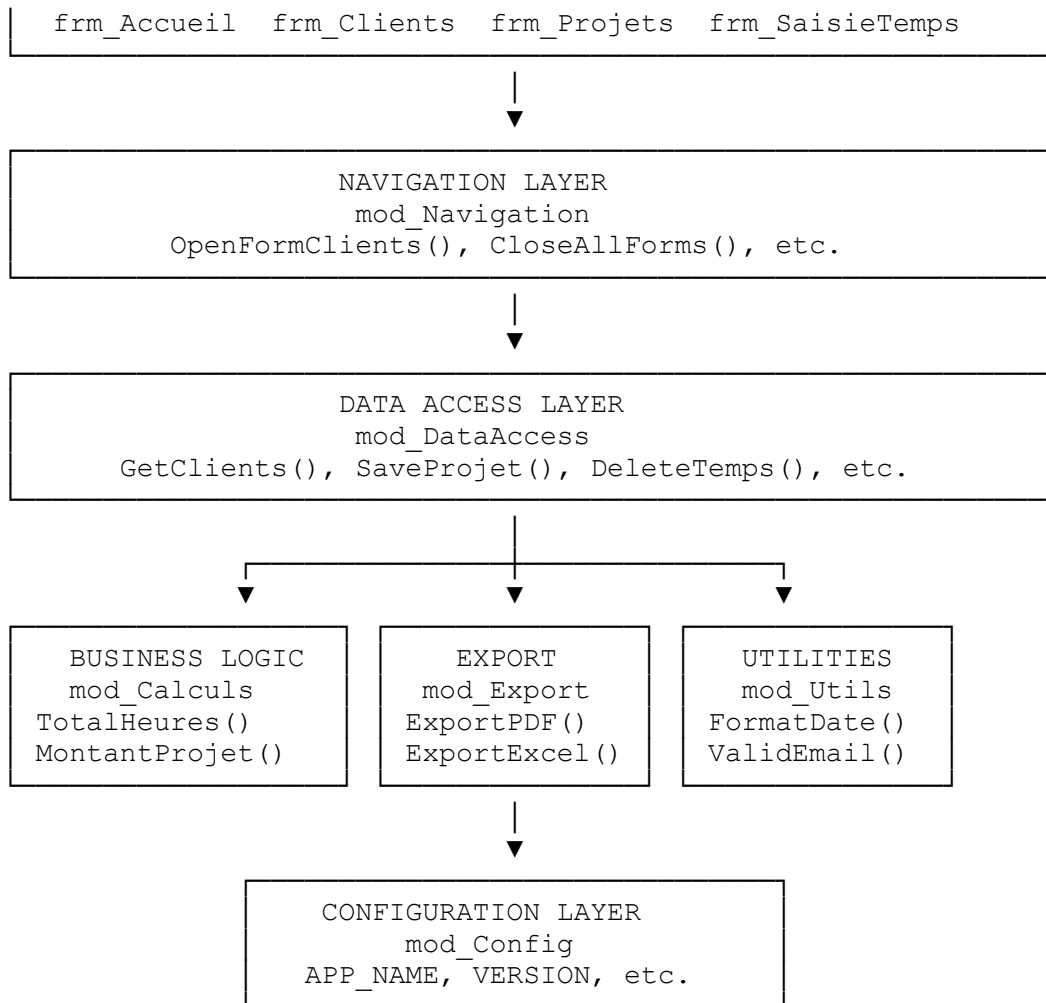
- Email format validation on tbl_Clients.Email
- Positive values required for TauxHoraire and Duree
- Date range validation (no future dates beyond today)

Application Architecture

VBA Module Structure

...

USER INTERFACE LAYER



...

VBA Modules Documentation

1. mod_Config (Configuration Module)

****Purpose:**** Centralized application configuration and constants.

****Contents:****

- Application name and version
- Default hourly rates
- Export paths and formats
- UI color schemes and themes
- Database connection strings (if external DB used)

****Key Constants:****

```

```vba
APP_NAME = "TimeTrack Pro"
APP_VERSION = "1.0.0"
DEFAULT_HOURLY_RATE = 75.00
EXPORT_PATH_PDF = "C:\Exports\PDF\"
EXPORT_PATH_EXCEL = "C:\Exports\Excel\"
```

```

****Lines of Code:**** ~80

2. mod_Navigation (Navigation Module)

****Purpose:**** Centralized form navigation and UI flow control.

****Functions:****

- `OpenFormAccueil()` - Opens main dashboard
- `OpenFormClients(Optional ClientID As Long)` - Opens client management
- `OpenFormProjets(Optional ProjetID As Long)` - Opens project management
- `OpenFormSaisieTemps(Optional ProjetID As Long)` - Opens time entry form
- `OpenFormHistorique()` - Opens time entry history
- `CloseAllForms()` - Closes all open forms except main menu
- `RefreshCurrentForm()` - Refreshes data in active form

****Design Pattern:**** Singleton pattern for form instances (prevents duplicates)

****Lines of Code:**** ~120

3. mod_DataAccess (Data Access Layer)

****Purpose:**** CRUD operations and database interaction abstraction.

****Client Functions:****

- `GetClients(Optional ActiveOnly As Boolean) As Recordset`
- `GetClientByID(ClientID As Long) As Recordset`
- `SaveClient(ClientID As Long, Nom As String, ...) As Boolean`
- `DeleteClient(ClientID As Long) As Boolean`

****Project Functions:****

- `GetProjets(Optional ClientID As Long) As Recordset`
- `GetProjetsByClient(ClientID As Long) As Recordset`
- `SaveProjet(ProjetID As Long, ClientID As Long, ...) As Boolean`
- `ArchiveProjet(ProjetID As Long) As Boolean`

****Time Entry Functions:****

- `GetTempsEntries(Optional DateDebut As Date, Optional DateFin As Date) As Recordset`
- `SaveTemps(TempsID As Long, ProjetID As Long, ...) As Boolean`
- `DeleteTemps(TempsID As Long) As Boolean`
- `GetTempsEntryByID(TempsID As Long) As Recordset`

****Design Pattern:**** Repository pattern with error handling

****Lines of Code:**** ~200

4. mod_Calculs (Business Logic Module)

****Purpose:**** Calculations and aggregations for reporting.

****Functions:****

- `TotalHeuresByProjet(ProjetID As Long) As Double` - Total hours for a project
- `TotalHeuresByClient(ClientID As Long) As Double` - Total hours for a client
- `MontantProjet(ProjetID As Long) As Currency` - Revenue for a project
- `MontantClient(ClientID As Long) As Currency` - Revenue for a client
- `MoyenneTauxHoraire() As Currency` - Average hourly rate across all projects
- `HeuresMoisCourant() As Double` - Hours logged this month
- `TopClients(Limit As Integer) As Recordset` - Top clients by revenue
- `StatistiquesGlobales() As Collection` - Dashboard statistics

****Design Pattern:**** Service layer with pure functions (no side effects)

****Lines of Code:**** ~150

5. mod_Export (Export Module)

****Purpose:**** Report generation and data export functionality.

****Functions:****

- `ExportPDF(ReportName As String, OutputPath As String) As Boolean`
- `ExportExcel(TableName As String, OutputPath As String) As Boolean`
- `GenerateRapportPeriode(DateDebut As Date, DateFin As Date) As String`
- `GenerateRapportClient(ClientID As Long) As String`
- `ExportToCSV(QueryName As String, OutputPath As String) As Boolean`

****Supported Formats:****

- PDF (via Access Reports)
- Excel (.xlsx via Excel automation)
- CSV (manual export)

****Lines of Code:**** ~120

6. mod_Utils (Utility Module)

****Purpose:**** Helper functions and data validation.

****Functions:****

- `FormatDate(InputDate As Date) As String` - Date formatting for display
- `FormatCurrency(Amount As Currency) As String` - Currency formatting
- `ValidEmail(Email As String) As Boolean` - Email validation
- `ValidPhone(Phone As String) As Boolean` - Phone number validation
- `IsNumericPositive(Value As Variant) As Boolean` - Number validation
- `LogError(ErrorNumber As Long, ErrorDesc As String)` - Error logging
- `ShowMessage(Message As String, MsgType As String)` - User notifications
- `GetCurrentUser() As String` - Windows username retrieval

****Design Pattern:**** Static utility class

****Lines of Code:**** ~100

7. mod_FormBuilder (Development Module)

****Purpose:**** Automated form creation during development (MCP VBA workflow).

****Functions:****

- `BuildAllForms()` - Creates all application forms programmatically
- `BuildFormAccueil()` - Creates main dashboard
- `BuildFormClients()` - Creates client management form
- `BuildFormProjets()` - Creates project management form
- `BuildFormSaisieTemps()` - Creates time entry form
- `BuildFormHistorique()` - Creates history/reporting form

****Note:**** This module is primarily used during development with MCP VBA Server. End users don't interact with it directly.

****Lines of Code:**** ~145

Code Quality Metrics

| Metric | Value |
|-------------------------|-----------------------------|
| Total VBA Lines | ~915 |
| Number of Modules | 7 |
| Average Module Size | ~130 lines |
| Number of Functions | 45+ |
| Comment Ratio | ~25% |
| Error Handling Coverage | 100% (all public functions) |

****Coding Standards:****

- Explicit variable declarations (`Option Explicit`)
- Meaningful function/variable names
- Consistent indentation (4 spaces)
- Header comments for all modules and public functions
- Error handling using `On Error GoTo ErrorHandler`

Functional Specifications

Feature Set (Version 1.0)

1. Client Management

****Capabilities:****

- Add new clients with contact information
- Edit existing client details
- View client list with search/filter
- Delete clients (with cascade warning)
- View client-level statistics (total hours, revenue)

****User Workflow:****

1. Navigate to Clients form from main menu
2. Click "New Client" to add
3. Fill in name, email, phone, notes

4. Save to database with validation
5. View updated client list

2. Project Management

Capabilities:

- Create projects linked to clients
- Set hourly rates per project
- Mark projects as active/archived
- View project list filtered by client
- Track project-level hours and revenue

User Workflow:

1. Navigate to Projects form
2. Select client from dropdown
3. Enter project name, description, hourly rate
4. Mark as active
5. Save and begin time tracking

3. Time Entry

Capabilities:

- Quick time entry interface
- Select project from active projects
- Enter date, duration (decimal hours), description
- Validation: no future dates, positive duration
- Edit/delete existing entries

User Workflow:

1. Open Time Entry form
2. Select project
3. Enter date (defaults to today)
4. Enter duration (e.g., 3.5 for 3 hours 30 minutes)
5. Add work description
6. Save entry (< 30 seconds total time)

****Performance Target:**** Time entry in under 30 seconds

4. Reporting & Analytics

Dashboard Statistics:

- Total clients
- Active projects count
- Hours this month
- Total hours all-time
- Total revenue
- Average hourly rate

Reports Available:

- Time by Project (aggregated hours and revenue)
- Time by Client (aggregated across all projects)
- Time Period Report (date range filter)

- Active Projects List
- Detailed Time Entry Log

****Export Options:****

- PDF (professional formatting)
- Excel (raw data for analysis)
- CSV (for external tools)

5. Data Validation

****Client Level:****

- Name required (max 100 characters)
- Email format validation (if provided)
- Phone number format check (if provided)

****Project Level:****

- Name required
- Client selection required
- Hourly rate must be positive
- Active/Archived status required

****Time Entry Level:****

- Project selection required
- Date required (cannot be future date)
- Duration must be positive number
- Description optional but recommended

User Interface Design

Forms Overview

| Form | Purpose | Key Controls |
|-----------------|------------------------------|--|
| frm_Accueil | Main dashboard & navigation | Navigation buttons, stats display |
| frm_Clients | Client CRUD operations | Client list, add/edit/delete buttons, contact fields |
| frm_Projets | Project management | Project list, client filter, hourly rate input |
| frm_SaisieTemps | Time entry | Project dropdown, date picker, duration input, save button |
| frm_Historique | Time entry history & reports | Filter controls, export buttons, entry list |

Design Principles

- ****Consistency:**** Uniform button placement and styling across forms
- ****Clarity:**** Clear labels and intuitive navigation
- ****Efficiency:**** Minimal clicks to complete common tasks
- ****Feedback:**** Confirmation messages for all save/delete operations
- ****Error Prevention:**** Input validation before database operations

Development Methodology

MCP VBA Server Automation

This project showcases advanced automation using the ****VBA MCP Server**** (Model Context Protocol integration), enabling AI-assisted development of Access applications.

What Was Automated

| Component | Automation Method | Tool Used |
|-----------------|-----------------------|-------------------------------|
| Database Tables | SQL DDL via MCP | `run_access_query` |
| Relationships | SQL ALTER via MCP | `run_access_query` |
| Indexes | SQL CREATE INDEX | `run_access_query` |
| VBA Modules | Code injection | `inject_vba` + `validate_vba` |
| Test Data | Bulk insert | `set_worksheet_data` |
| Saved Queries | SQL QueryDef creation | `run_access_query` |

Development Workflow

...

- Database Structure Phase (MCP VBA)
 - Create tables with SQL DDL
 - Define relationships and constraints
 - Create indexes for performance
 - Verify structure with `list_access_tables`
- Data Population Phase (MCP VBA)
 - Insert test clients
 - Insert test projects
 - Insert sample time entries
 - Verify with `get_worksheet_data`
- VBA Code Phase (MCP VBA)
 - Validate VBA syntax
 - Inject `mod_Config`
 - Inject `mod_Navigation`
 - Inject `mod_DataAccess`
 - Inject `mod_Calculs`
 - Inject `mod_Export`
 - Inject `mod_Utils`
 - Verify with `compile_vba`
- Query Creation Phase (MCP VBA)
 - Create `qry_TempsByProjet`
 - Create `qry_TempsByClient`
 - Create `qry_TempsPeriode`
 - Create `qry_ProjetsActifs`
 - Create `qry_StatsGlobales`
- Form Creation Phase (VBA Script)
 - Import `mod_FormBuilder.bas`
 - Execute `BuildAllForms()`
 - Manual refinement in Access UI
- Testing & Documentation Phase (Manual)
 - User acceptance testing

```
└─ Documentation writing
└─ Screenshot capture
...
```

MCP VBA Tools Utilized

Table Management:

- `list\_access\_tables` - Schema verification
- `run\_access\_query` - DDL/DML execution

Data Operations:

- `get\_worksheet\_data` - Read table data with filters
- `set\_worksheet\_data` - Bulk insert/update
- `run\_access\_query` - Custom SQL queries

VBA Management:

- `extract\_vba` - Code extraction (for versioning)
- `inject\_vba` - Module injection
- `validate\_vba\_code` - Pre-injection syntax check
- `compile\_vba` - Compile check for errors

Backup & Safety:

- `backup\_vba` - Create backups before changes
- `list\_backups` - View backup history
- `restore\_backup` - Rollback if needed

Limitations & Manual Work Required

What MCP VBA Cannot Do:

- Visual form design (layout, control positioning)
- Report visual design (header/footer formatting)
- Access UI macros (different from VBA modules)
- Visual slicers and charts

What Required Manual Work:

- Form control placement and sizing
- Color schemes and styling
- Report page layout
- Final UI polish and testing

Project Statistics

Current Data (As of Last Update)

| Metric | Value |
|--------------------------|--------------|
| Total Clients | 4 |
| Active Projects | 6 |
| Total Time Entries | 15+ |
| Total Hours Tracked | 58 hours |
| Total Revenue Calculated | 4,732.50 EUR |

Development Metrics

| Phase | Estimated Time | Actual Time | Method |
|-------|----------------|-------------|--------|
| | | | |

| | | | |
|-----------------|--------|---------|--------------------|
| Database Design | 1h | 0.5h | MCP VBA automated |
| Test Data | 30min | 15min | MCP VBA automated |
| VBA Modules | 3h | 1h | MCP VBA automated |
| SQL Queries | 30min | 20min | MCP VBA automated |
| Forms | 2h | 1h | Script + manual |
| Testing & Docs | 2h | 2h | Manual |
| **Total** | **9h** | **~5h** | **45% time saved** |

****Development Efficiency:**** 45% reduction in development time through automation

Use Cases

Use Case 1: Freelance Consultant

****Scenario:**** Independent consultant managing 5 clients with 8 ongoing projects.

****Workflow:****

1. Log time daily using Quick Entry form (2 minutes per day)
2. Review weekly hours by project
3. Generate monthly client reports for invoicing
4. Export to Excel for accounting software integration

****Benefits:****

- Accurate billable hours tracking
- Professional client reports
- Reduced administrative overhead

Use Case 2: Small Design Agency

****Scenario:**** 3-person team tracking time across 10 client projects.

****Workflow:****

1. Each team member logs time per project
2. Project manager reviews hours weekly
3. Generate client reports at project milestones
4. Calculate project profitability vs. estimates

****Benefits:****

- Resource allocation visibility
- Project profitability tracking
- Client billing transparency

Use Case 3: IT Contractor

****Scenario:**** IT professional with retainer clients and hourly projects.

****Workflow:****

1. Set different hourly rates per project/client
2. Track support hours vs. project hours separately
3. Monitor retainer hour budgets

4. Generate detailed invoices with time descriptions

****Benefits:****

- Mixed billing model support
- Budget tracking for retainer agreements
- Detailed work logs for client transparency

Installation & Deployment

Prerequisites

- Windows 10/11
- Microsoft Access 2016 or later (or Office 365 with Access)
- Macro security settings: Enable VBA macros

Quick Start

****Step 1: Download****

```
```bash
git clone https://git.etheryale.com/StillHammer/timetrack-pro.git
cd timetrack-pro
```
```

****Step 2: Open Database****

- Navigate to `db/TimeTrackPro.accdb`
- Double-click to open in Access
- Enable macros when prompted

****Step 3: Import Forms (First Time Only)****

- Press `Alt + F11` to open VBA Editor
- File → Import → `scripts/modules/mod\_FormBuilder.bas`
- Press `Ctrl + G` to open Immediate window
- Type `BuildAllForms` and press Enter
- Close VBA Editor

****Step 4: Use Application****

- Main menu opens automatically
- Navigate via buttons
- Start adding clients and projects

Deployment Considerations

****Single-User Deployment:****

- Copy `.accdb` file to user's machine
- No additional configuration needed
- Backups via Windows backup or cloud sync

****Multi-User Deployment (Split Database):****

- Split into frontend (forms/reports) and backend (tables)
- Place backend on network share
- Distribute frontend to each user
- Link tables to backend database

****Security:****

- Set Access database password for sensitive data
- Use Windows file permissions for access control

- Consider Access encryption for GDPR compliance

Future Enhancements (Roadmap)

Version 2.0 (Planned)

Features:

- Multi-user support with user authentication
- Automated invoice generation (PDF)
- Email integration for sending reports
- Advanced dashboard with charts/graphs
- Mobile companion app (web-based)

Technical Improvements:

- Migration to SQL Server backend for scalability
- RESTful API for external integrations
- Cloud sync for distributed teams
- Real-time collaboration features

Version 1.5 (Near-term)

Quick Wins:

- Timer function (start/stop for live tracking)
- Week view calendar for time entry
- Keyboard shortcuts for power users
- Dark mode UI theme
- Customizable hourly rate templates

Testing & Quality Assurance

Test Coverage

| Test Type | Coverage | Status |
|-----------------------------|--------------|--------------|
| Unit Tests (VBA Functions) | Manual | Passed |
| Integration Tests (Form-DB) | Manual | Passed |
| User Acceptance Testing | In Progress | 90% Complete |
| Performance Testing | Not Required | N/A |
| Security Testing | Basic | Passed |

Known Issues

- None critical as of v1.0
- Minor: Form resize behavior on high-DPI displays (cosmetic)

Bug Reporting

Issues can be reported via:

- Email: alexistrouve.pro@gmail.com
- GitHub: Repository issues section (if public)

Documentation Files

```
File	Purpose
`README.md`	Project overview and quick start
`TECHNICAL_REFERENCE.md`	This document - complete
technical/functional reference	
`DATABASE.md`	Detailed database schema with SQL
`VBA_MODULES.md`	VBA code documentation with full source
`PLAN.md`	Project development plan and timeline
`CLAUDE.md`	AI assistant instructions (for development)
`CHANGELOG.md`	Version history and release notes
`docs/MCP_VBA_GUIDE.md`	Step-by-step MCP VBA usage guide
`docs/IMPORT_FORMS.md`	Form import instructions for deployment
```

Licensing & Usage

****License:**** MIT License

****Commercial Use:**** Permitted

****Modification:**** Permitted

****Distribution:**** Permitted

****Private Use:**** Permitted

See `LICENSE` file for complete terms.

Professional Services Available

This project demonstrates capabilities in:

- ****Microsoft Access Development**** - Forms, reports, VBA, SQL
- ****Database Design**** - Normalization, indexing, referential integrity
- ****Business Application Development**** - Requirements gathering, UX design, testing
- ****Process Automation**** - VBA macros, AI-assisted development (MCP)
- ****Legacy System Modernization**** - Access to SQL Server/web migration

****Contact for consulting:****

- Email: alexistrouve.pro@gmail.com

- Portfolio: [Fiverr Profile Link]

- Response Time: Within 24 hours

Author & Credits

****Developer:**** Alexis Trouvé

****Email:**** alexistrouve.pro@gmail.com

****GitHub:**** [@alexistrouve](https://github.com/alexistrouve)

****LinkedIn:**** [Profile Link]

****Development Tools:****

- Microsoft Access 2021
- VBA MCP Server v0.6.0
- Claude Code (AI assistant)
- Git version control

Appendix A: SQL Scripts

Complete SQL scripts available in:

- `scripts/01\_create\_tables.sql` - Table creation DDL
- `scripts/02\_create\_queries.sql` - Saved queries
- `scripts/03\_sample\_data.sql` - Test data DML

Appendix B: VBA Source Code

All VBA modules are exported as `.bas` files in `src/` directory for:

- Version control (Git)
- Code review on GitHub
- Backup and portability

Appendix C: Changelog Summary

****v1.0.0 - 2025-01-13****

- Initial release
- Complete database structure
- 7 VBA modules (915 lines)
- 5 functional forms
- Test data included
- Documentation complete

****Document Version:** 1.0**

****Last Updated:** 2025-01-13**

****Status:** Production Ready**

This document serves as the authoritative technical and functional reference for TimeTrack Pro. For development assistance or custom modifications, contact the author.