**Freelance-Force-Suite Project – Full Explanation**

# Phase 1: Problem Understanding & Industry Analysis

👉 Goal: Understand what we’re building and why.

## 1. Requirement Gathering

* + Manage all client information and project details in one central location.
  + Track billable hours accurately against specific projects and tasks.
  + Automate the creation of professional invoices based on logged time.
  + Generate reports to analyze revenue and project profitability.

## 2. Stakeholder Analysis

## 

○ **The Freelancer** (Needs a simple, all-in-one system to manage their entire business lifecycle, from client acquisition to getting paid. This stakeholder acts as the Admin, End User, and Manager.).

## 3. Business Process Mapping

* + **Current Process:** A fragmented workflow using multiple tools: a spreadsheet for client tracking, a separate app for task management (like Trello), another for time tracking, and a word processor for manually creating invoices. This process is time-consuming and prone to data entry errors.
  + **Future Process:** A client (Account) has a Project\_\_c record created. The freelancer logs time against the project using Time\_Log\_\_c records. At the end of a billing cycle, a button click automatically generates an Invoice\_\_c record, summing up all unbilled hours. The freelancer then manually updates the invoice status to "Paid" upon receiving payment.

## 4. Industry-specific Use Case Analysis

* Freelancers often juggle multiple clients and projects simultaneously. The system needs to provide a clear, consolidated view of all ongoing work and outstanding payments. It must also handle both fixed-price projects and hourly billing models.

## 5. AppExchange Exploration

* **Decision for Payment Tracking:** To ensure the project remains **completely cost-free**, no paid payment processing integrations (like Stripe or PayPal connectors) will be used. The core business need will be met by creating a custom Invoice Status field, which the freelancer will manually update from "Sent" to "Paid" after receiving funds through their existing, external methods (e.g., bank transfer).

# Phase 2: Org Setup & Configuration

👉 Goal: Prepare Salesforce environment.

1. **Salesforce Editions**

○ A free **Developer Edition Org** has been created and is in use for all development and configuration.

## 2. Company Profile Setup

○ The freelancer's business profile, including a business name, address, default time zone, and primary currency, has been configured in Setup > Company Information.

## 3. Business Hours & Holidays

○ Standard business hours (e.g., 9:00 AM - 6:00 PM, Monday-Friday) have been defined to ensure any time-based automations run correctly.

1. **Fiscal Year Settings**

○ A **Standard Fiscal Year** (January-December) has been configured to simplify revenue reporting for the calendar year.

1. **User Setup & Licenses**

○ A single user record for the freelancer has been configured with a **Salesforce** license.

## 6. Profiles

○ The user is assigned the **System Administrator** profile, providing full access to build and customize the application.

7. **Roles**

○ A top-level role (e.g., "Freelancer") has been created and assigned to the user as a best practice for reporting and future scalability.

## 8. OWD (Org-Wide Defaults)

○ The Organization-Wide Defaults for all key standard objects (Account, Opportunity) and custom objects (Project\_\_c) have been set to **Private** to ensure maximum data security.

1. **Login Access Policies**

* + The default login and password policies for the org have been reviewed and confirmed.

## 10. Deployment Basics

* + The Developer Org has been successfully connected to a local VS Code project. All setup and configuration from this phase have been retrieved and pushed to a public GitHub repository for version control.

# Phase 3 : Data Modeling & Relationships

👉 Goal: Build the core data structure to manage clients, projects, time logs, and invoices.

1. **Standard & Custom Objects**

* + **Standard Objects:**
* Account: To store client company information.
* Contact: To store individual client contact details.
  + **Custom Objects:**
* Project\_\_c: The central object to track all details of a single project for a client.
* Time\_Log\_\_c: To record individual blocks of billable time worked on a project.
* Invoice\_\_c: To generate and track the status of invoices sent to clients.

.

## 2. Fields

* **Project Object Fields :**
* Status (Picklist): Tracks the application's current stage (e.g., Planning, In Progress, Completed, On Hold).
* Hourly Rate (Currency): The agreed-upon billing rate for the project.
* Start Date (Date): When the project officially begins.
* End Date (Date): The projected or actual completion date
* **Time Log Object Fields :**
* Date (Date): The date the work was performed.
* Hours (Number): The number of hours worked (e.g., 2.5).
* Description (Long Text Area): A summary of the work completed.
* Status (Picklist): "Unbilled" or "Billed". Defaults to "Unbilled".

* **Invoice Object Fields :**
* Status (Picklist): Tracks the invoice's lifecycle (e.g., Draft, Sent, Paid, Overdue).
* Due Date (Date): The date payment is due.
* Total Amount (Currency): The total calculated amount of the invoice.

## 3. Record Types

* Record types are not required for the initial build of this project, as the process for all projects is the same. They could be added later if the freelancer decides to offer different types of services (e.g., "Fixed Price Project" vs. "Monthly Retainer").

1. **Page Layouts**

* The Project page layout is the main workspace. It has been configured to show the Time Logs and Invoices as related lists, so the freelancer can see all financial details for a project in one place.

1. **Compact Layouts**

## The Project compact layout is configured to show the Project Name, Client Name, and Status at a glance on mobile and in list views.

## Schema Builder

## 

* The Schema Builder was used to visually design and confirm the relationships between Accounts, Contacts, Projects, Time Logs, and Invoices, ensuring the data model is logical and scalable.

1. **Lookup vs. Master-Detail vs. Hierarchical Relationships**

* **Project to Account:** A required Lookup Relationship. A project must be linked to a client account, but they are independent records.
* **Invoice to Project:** A required Lookup Relationship. An invoice must be related to a project.
* **Time Log to Project:** A **Master-Detail Relationship**. A time log is a direct child of a project. If a project is deleted, all its associated time logs are automatically deleted. This ensures data integrity and allows for roll-up summary calculations

## 8. External Objects

* External objects are not in scope for this project. They could be used in a future enhancement to connect Salesforce with an external accounting system without migrating data.