

Goal:

My goal is to solve the problem faced by freelance developers in managing and organizing different elements of their work. This includes various customers, numerous projects, associated tasks, and related invoices. Without a centralized system, maintaining all these aspects can be time-consuming, prone to error, and generally inefficient, potentially impeding a freelancer's productivity and effectiveness.

Requirements Document

The freelance developers require a solution that allows them to:

1. Register and manage their profile with basic details (Full_Name, Email, Password, Phone_Number, Username).
2. Manage a list of customers, storing information such as Name, Contact_Email, and Billing_Address.
3. Generate and manage invoices tied to a customer, tracking critical parameters like Date_Issued, Due_Date, Total_Amount, and Status.
4. Create and manage projects, inclusive of details like Title, Brief, Start_Date, End_Date, Budget, and Status, each linked to a specific customer.
5. Assign and track tasks within projects with detailed information like Title, Description, Start_Date, Due_Date, and Status.

Nouns Identified:

User, Full_Name, Email, Password, Phone_Number, Username, Customer, Customer ID, Name, Contact_Email, Billing_Address, Invoice, Invoice_ID, Date_Issued, Due_Date, Total_Amount, Status, Project, Project_ID, Title, Brief, Start_Date, End_Date, Budget, Task, Task_ID, Description.

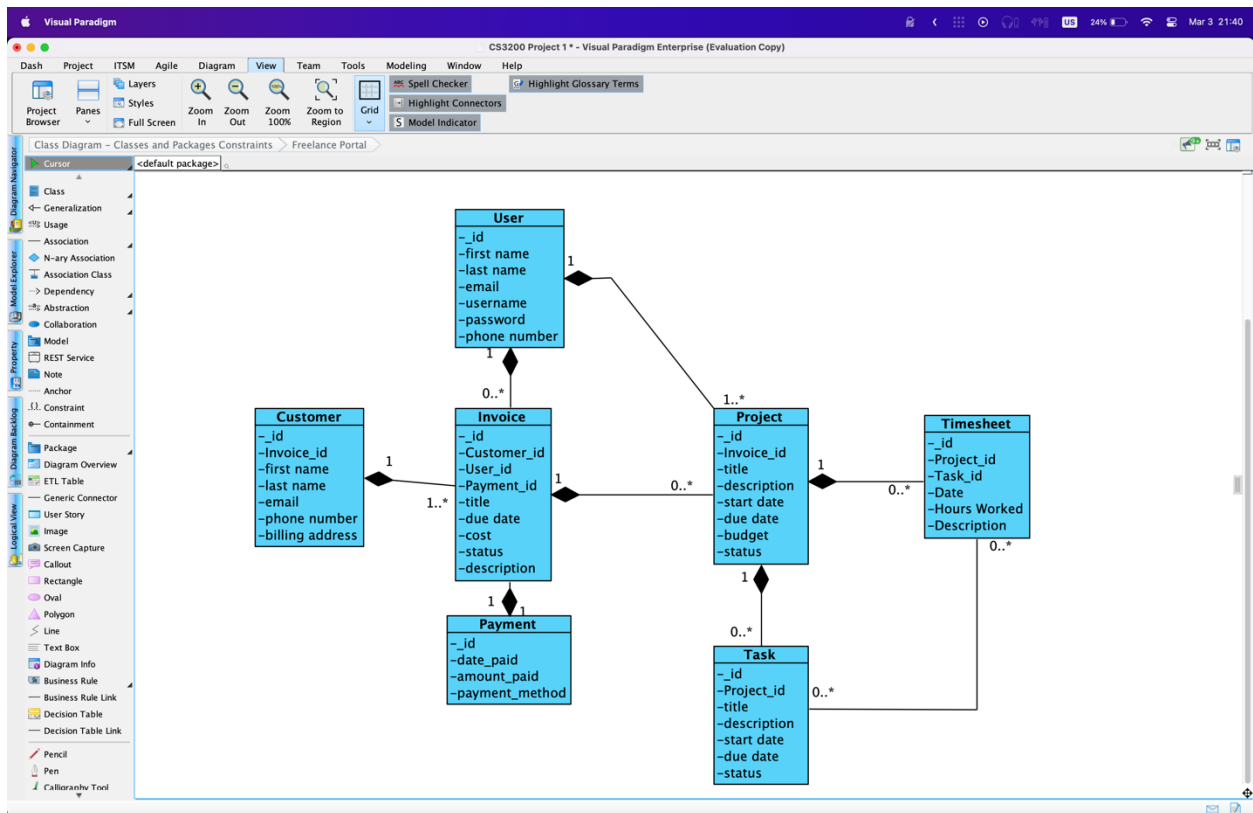
Actions Identified:

Register, Manage Profile, Manage Customer List, Store Information, Generate Invoice, Track Invoice, Create Project, Manage Project, Assign Task, Track Task, Update Status.

Redis Functionality:

- User session tracker
 - A tracker that creates a unique session token for an individual user. The token would be set to expire after X amount of time. Once the time has passed the user would automatically be logged out of the session/web app. The app would query the redis every minute or so to check if the key has expired.
 - Simple string e.g. sessionId:userId -> value based on time of login
 - Commands
 - Create
 - SET sessionId:userId new Date().getTime()
 - EXPIRE sessionId:userId 3600
 - Read
 - GET sessionId:userId
- Users' most recent project
 - Caches the id of the most recent project the user accessed. It would also cache the important data of the project so that the user does not have to wait to access it. The most recent project would be displayed to the user for faster access. Updates once a user opens a project.
 - Hash e.g. recentProject:userId -> project_name, project_description, project_start_date, project_end_date, project_budget, project_status
 - Commands
 - Create – HSET recentProject:userId "project_name"
 - Read – HGETALL recentProject:userId
 - Update – HSET recentProject:userId "project_name"
 - Delete – HDEL recentProject:userId
- User's frequently accessed projects
 - Finds and tracks the projects frequently accessed and caches them in order of frequency.
 - Sorted Set e.g. frequentProjects:userId
 - Commands

- Create – ZADD frequentProjects:userId -> timesAccessed, projectId
- Read – ZRANGE frequentProjects:userId -1 10 BYSCORE REV
- Whenever a user accesses a project
 - Update – ZINCRBY frequentProject:userId 1 projectId
- Delete DEL frequentProject:userId



user	
_id: ObjectID	
last_name: STRING	
first_name: STRING	
email: STRING	
username: STRING	
password: STRING	
phone_number: STRING	

customer	
_id: ObjectID	
first_name: STRING	
last_name: STRING	
email: STRING	
phone_number: STRING	
billing_address: STRING	

invoice	
_id: ObjectID	
invoice_title: STRING	
invoice_due_date: DATE	
invoice_cost: FLOAT	
invoice_description: STRING	
invoice_status: STRING	
user_id: ObjectID	
customer_id: ObjectID	

projects	
project_name: STRING	
project_description: STRING	
project_start_date: DATE	
project_end_date: DATE	
project_budget: FLOAT	
project_status: STRING	

payment	
date_paid: DATE	
amount_paid: FLOAT	
payment_method: STRING	

tasks	
task_name: STRING	
task_description: STRING	
task_start_date: DATE	
task_end_date: DATE	
task_status: STRING	

timesheets	
timesheet_date: DATE	
timesheet_hours_worked: INT	
timesheet_description: STRING	

