

**TREIS ADIUTOR SYSTEM: A WEB-BASED MULTI-STAKEHOLDER
PLATFORM FOR AUTOMATED SERVICE REQUEST PROCESSING,
PROJECT TASK MANAGEMENT, AND INTEGRATED FINANCIAL
OPERATIONS**

A Final Project

Presented to the

Faculty of College of Computer Studies

Laguna State Polytechnic University

Santa Cruz Campus

In partial Fulfilment of the Requirements for the course

CLIENT SERVER TECHNOLOGIES

Submitted by:

Princess Anne B. Azucena
Sofia Lorraine Gonzaga
Ma. Lena Therese S. Quizon
Mark Andrew Z. Soliman

Submitted to:

Harlene Gabrielle E. Origines
Course Instructor

December 2025

CHAPTER I

INTRODUCTION

Technological advancements are rapidly progressing worldwide, driving organizations to adopt innovative solutions to maintain competitive edge. Modern project management relies on a web-based, centralized platform with automation, real-time communication, and online financial solutions to improve workflow, collaboration, and project execution. Globally, modern digital tools are implemented by many organizations that standardize project management, enhance communication among stakeholders, and provide secure and reliable transaction management.

The Treis Adiutor Web-Based Multi-Stakeholder System for Automated Service Request Processing, Project Task Management, and Integrated Financial Operations was developed as a unified platform that facilitates project management between different users. It provides an efficient service and gives coherent collaboration among the clients, adiutors (service providers), and administrators with the support of end-to-end workflows. The client users are allowed to submit a wide range of projects personalized to their choices and preferences. To effectively complete the project's requests, the adiutors or service providers will execute the projects with full attention to the functionality of every feature involved. Meanwhile, the administrator oversees the overall project management to assure proper execution of tasks. The system offers various types of digital and technological development and services. As technology and innovation have evolved, this project utilizes a centralized digital platform for

automating tasks, monitoring progress, and real-time communication while ensuring organized and reliable project management outcomes.

The main problem and challenges that Treis Adiutor System (TAS) aims to uncover are inefficiencies and inconsistencies that result in unreliable outcomes. The existing method of Treis Adiutor could result in redundancy of work, slower services, difficulty in tracking project processes, and poor integration of payments and management systems. These challenges highlight the need for a centralized platform solution that aligns user-friendly project management with modern technological standard methods.

To address these challenges, the system proposes a Web-Based Multi-Stakeholder System for Automated Service Request Processing, Project Task Management, and Integrated Financial Operations. The proposed system is relevant as it aligns the need for technological practices with standardized features that will address different technological gaps within the organizations. This project will contribute by providing different Application Programming Interfaces (APIs) with automated workflows, Real-Time Communication, Payment Solution, and Time Tracking and Productivity.

The demand for digital innovation and transformation for operations becomes more essential in various industries. Project Management processes significantly evolve for both local and international approaches with the advancement of technologies. Despite the tools' availability in local settings, many businesses still rely on manual work and use separate systems for their project monitoring and management which leads to inefficiency, delay, and inconsistency.

Whereas, international organizations implement a centralized digital platform conducting Artificial Intelligence (AI), Standardized Data Storage Systems, Project Monitoring, and Enhanced Integration of Communication, and Payment Solutions. These features guarantee automated project management and seamless collaboration between stakeholders. In essence, the proposed system Treis Adiutor System (TAS) is a localized implementation with global technological standards, designed to provide the tools needed for managing projects, monitor progress, and maintain smooth transactions — all modern project management systems worldwide.

Background of the Study

Client-server technology is a fundamental model in modern software systems, allowing devices or applications (clients) to access services and resources from powerful centralized servers. This architecture is the backbone of many web-based platforms because it divides the responsibilities of the user interface side and the server side. This model improves efficiency by assigning duties appropriately to the client and server. Client-side handles all user interaction, while servers process, manage, and secure data. Meanwhile, the server side is assigned for processing the requests, managing and securing data, implementing system protocols, and ensuring that all operations run smoothly. On top of that, it enhances the over-all system performance because it provides consistency, accuracy, and security of the data.

Client-server technology is essential for Treis Adiutor System, it ensures that interaction between client, Adiutors, and Administrator (Admin) is seamless

and efficient. This model (client-server technology) helps the server-side of the system to manage user information, process service requests, manage projects, handle payment, store documents, and deliver real-time updates. In contrast to the server-side, the client-side is responsible for presenting all the information and contents, like the user's dashboard that are visually appealing, clean, minimalist and has a design pattern for better experience for the user. Furthermore, the model supports responsiveness, which makes it accessible to users using the system from various locations or devices and still receive error-free and exact data. Client-server technology makes TAS more protected, safe, and able to automate complex processes, by the end it solves the problems with common project management and enhances cooperation among all system users.

Research Problem

The standardized method in technology is progressively emerging in numerous industries with project management being a significant area of influence to boost and enhance the execution and performance of development. By utilizing modern technologies, these allow us to manage repetitive tasks and allocate resources effectively resulting in reliable and improved project execution and management. The manual practices of Treis Adiutor can bring errors, poor project progress tracking, slower services, and difficulty in financial transactions. The need for meaningful understanding and deliberate investigation to overcome these barriers becomes paramount for sustained growth and innovation in the current project management workflow of Treis Adiutor.

Generally, the problem of this research is the absence of automated and standardized project management platforms. The proposed system will not only enhance the management of projects but also enable service providers to concentrate on strategic tasks, ultimately providing administrators with an organized and comprehensive solution for overseeing projects within the project management sector.

1. How does centralized TAS can improve the project workflows compared to the existing manual process of Treis Adiutor?
2. How does the digital platform of TAS contribute to the project execution, communication, scheduling, and payment transactions within the workflow of Treis Adiutor?
3. What features are necessary to ensure that TAS delivers a secure and reliable environment for managing and executing project tasks?

The cause of the identified problem from the existing method relies on manual processes and non-integrated systems to track projects and resources, which often are inefficient and prone to errors. The absence of centralized tools increases project delays and cost overruns. TAS, however, monitors the tasks progress and ensures the works are properly organized.

A study titled "*Enhancing Project Management Success through Artificial Intelligence* (Roman Reznikov, January 2025.) emphasized that automation of project management tools has introduced a paradigm shift, improving project execution and management by increasing efficiency, accuracy, and consistency.

The study reveals that AI enabled significantly reduces errors, delays, and manual workload. This demonstrates efficiency improvements by 50% streamlines communication and reporting, project delays by 35%. Reduces rework and issues by 50%, and improves utilization and productivity by 30-50%. This proves that manual project execution is more prone to poor tracking and communication, redundancy, and coordination issues, reinforcing the need for a standardized automated system.

The intended solution is the unified platform that monitors various types of project execution offering a modernized look on maximizing project success. The system will streamline task management, and give seamless collaboration shaping the future of project management paving the way for more successful project delivery in an increasingly competitive industry.

Project Objectives

The ultimate goal of Project Execution and Management System is to provide a simplified workflow, enhance collaboration, and improve productivity while reducing project delays and to bring improvements for the organization. This approach not only to keep on track but to deliver quality project results leading to consistent team success.

Specifically, this study seeks to:

1. **Improve project workflow** by replacing manual processes into a unified platform where all users can work and communicate together.

The system also supports real-time monitoring, task management, and resources allocation.

2. **Supports security and authentication operations.** The system features implement various APIs to secure data access and controlled possible attacks.
3. **Enhance project execution and collaboration.** The centralized system provides effective communication, scheduling, and documentation across the project tracking and performances.

Scope and Limitations of the Research

This section will show the overall coverage and limitations of the study, it specifies the focus of the research by defining its subject matter, objectives, setting, and time frame. Furthermore, it provides clarity to the limitations that may affect the factors surpassing the researcher's jurisdiction.

The system scope provides user roles: Client, Adiutor, and the Administrator. These three roles have different responsibilities to maintain the system's functionality. The system also covers the overall workflow of project execution that starts from service request submission to project completion. To lengthen this information, the system will be focusing on managing the service request, project management, task creation and assignment, progress monitoring which allows the client to see the progress of her project, and also create feedback for the work of the Adiutor.

Treis Adiutor System (TAS) also automates processes such as notification where the users can see updates about a project's progress and payment reminders. In addition to the processes, it includes payment processing through Maya, scheduling using Google calendar, and approval workflows. TAS utilizes Cloudflare R2 for cloud-based storage for files and documents for more accurate, secure and real-time data. Regarding the APIs, TAS implemented Firebase, Maya for payment, Brevo, Zoom for online meeting, Google Calendar for scheduling, Gemini AI for chatbot support. These enhanced the system's capability and functionality.

TAS offers functional coverage, including user authentication and account management for user's privacy and safety, clients are allowed to submit service requests and review or edit the request anytime. TAS supports flexible project payment methods such as milestones, downpayment, and one-time payment. It also offers status tracking and comprehensive task management including creation, assignment, and monitoring. That will be used in creating real-time notification and the use of communication space. Lastly, TAS has an analytics dashboard for monitoring and feedback that states the rate of the client's satisfaction towards a service.

The system is built accessible on desktop and mobile browsers, with the use of PHP, Laravel, Tailwind, Alpine.js, and MySQL database. TAS is intended for organizations or users within the project's target environment (e.g., academic, business, and service-oriented industry).

Thereafter, the coverage or scope of the TAS is followed by the limitations and constraints. The boundaries can affect the development and performance of the system. These boundaries include technical, user-related, and system integration factors that are inevitable. Identifying these constraints helps to address the capability of the system and set realistic expectations for its use and evaluation.

First in the list of constraints is the technical limitation. The system requires a stable network since it is using cloud-based programs. The performance of TAS also depends on server load or third-party API availability, which means that if a conflict happens there, the system might run slower, or the user might experience delays and errors. Secondly the integration limitations, the payment transactions only rely on Maya Business API other than that there's no other option. Gemini AI chatbot depends on the available data in the system meaning it may not always provide accurate responses.

Significance of the Study

The integration of different APIs offers significance as it enhances efficiency, decision-making, and outcomes. Particularly in organizations that continuously rely on non-integrated system and manual processes to execute and deliver the project. Additionally, the Tries Auditor can provide valuable insights into project management, enabling users to adapt to the challenging circumstances.

The system is significant to the following stakeholders:

1. The Treis Adiutor (Organization)

The integration of multiple API and modernized features of this system will benefit the Treis Adiutor as it enhances project operational efficiency, decision-making and provides reliable project outcomes. The traditional practices that TAS, held often experienced inconsistent results and delays. The system addresses these challenges by providing a centralized platform that automates service request, task management, and financial operations. As a result, this organization improved project operation, reduced redundancy, and worked for more reliable project execution.

2. The Service Providers (Adiutors)

The system offers valuable support in real-time access to project information, task assignments, and progress updates, enabling adiutors to effectively work for challenging project requirements. This also helps them to reduce manual workload and improves coordination. TAS enhances productivity, and accountability among service providers.

3. The Administrators

The system provides administrators with an organized platform that oversees the project operations including the project payment transactions. Through the centralized monitoring, and financial management,

administrators can ensure that the workflow properly executes and gives transparency. This enhances administration control and strengthens project governance.

4. The Businesses with Similar Organization Structures

The businesses with similar organization structures will gain valuable insights into the implementation of a web-based project management system. This study demonstrates how a centralized platform can improve project coordination, task management, and financial operations within multi-stakeholder environments. This will contribute to providing a knowledgeable centralized system. The findings may serve as a practical reference for organizations seeking to modernize their project workflows, enhance scalability, and adopt standardized technological solutions to improve operational efficiency and service delivery.

5. The Future System Developers

The system offers user-friendly features and guidance for integrating multiple APIs, implementing automation, time tracking, security and authentication, and reliable project management systems. The system architecture provides insights into developing user-centered solutions that align with standardized technology and organization needs.

The results of this study are expected to contribute to the understanding of how a centralized platform can optimize project

management, scheduling, and communication. This study holds the potential practices in local organization, academic institution, and future developers seeking to adopt the technological standard. TAS can effectively streamline project management and drive organizational success in diverse industries.

CHAPTER II

SYSTEM ANALYSIS AND DESIGN

Overview of Client-Server Model

Treis Adiutor System (TAS) is using the Client-Server Model, in this model the system is divided into two major components: the client, which is the user interface, and the server, which processes data, handles logic, and manages the storage. These two components are connected and communicated through the internet to deliver seamless processes.

Client side refers to the look, visual content, and functions inside the system that are used by the Clients, Adiutor and Administrators. Client-side technologies are used in the system to make it responsive and visible. In front-end technologies TAS used Tailwind CSS for UI design and layout, Alpine.js for interactive components, and Chart.js for data visualizations and analytics. Regardless, the client is responsible for sending requests to the server whenever the user tries to commit actions with the system (e.g., submitting user credential on login page, client submitting project request, uploading files for a project). Furthermore, the client displays information received from the server, and provides the requested interfaces for users to interact with this data (e.g., login page, user dashboard, client projects page, request project form, message page, and etc.)

The server side is the backend engine of the system that is responsible for processing business logic, storing or retrieving data, managing authentication, handling integrations with the external APIs, and ensuring secure and accurate transactions. The back-end technologies used in TAS are Laravel 12.33.0 and PHP 8.2.12 as the core back-end framework, MySQL for cloud database, Cloudflare R2 for storing files and data, Apache/Nginx on Windows Server as the deployment platform. These technologies are the key factor to make the system functional and reliable. Overall, the server ensures the system is functioning consistently even if multiple clients are interacting with the system at the same time.

Here's an example of the workflow from client to server (Service Request Submission: client fills out request forms that triggers the reCAPTCHA, request is sent to the server, server validates data and saves request to MySQL, server sends email notification via Brevo, server triggers admin dashboard and FCM notification, client updates UI showing "Request Submitted").

The benefits of Client-Server Architecture in TAS are it ensures security by minimizing exposure of sensitive operations (e.g., payments, approval). It also allows the system to expand its scalability in client and server side. Lastly it ensures the maintainability, reliability of data and consistency.

System Requirements

The project requirement outlines the application of project management tools used for development, deployment, and operation of the TAS. As a centralized platform consisting of an integrated API. This requires stable materials to perform a consistent system across clients, auditors (service providers), and the administrator. The following requirements are essential for both software and hardware requirements.

1. The Software Requirements

Operating System is designed for deployment, this includes:

- Windows 10 or Windows 11
- Windows Server 2019 or 2022 for production

Web Server

- Internet Information Services with the minimum version of IIS 10+, Apache Server (2.4+), and Nginx (1.18+).

PHP Environment

The PHP application requires **PHP version 8.2+** with the following extensions:

- BCMath, Ctype, Fileinfo, JSON, Mbstring, OpenSSL, PDO, Tokenizer, and XML

- **Composer:** Latest stable version for PHP dependency management
(install via Windows installer)

Frontend Build Tools

The following are required for front-end asset compilation.

- **Node.js 18+**
- **Npm package manager**

Database Management System

- **SQLite for development.**
- MySQL 8.0+ or PostgreSQL 13+ **for production.**

Security Requirement

- The production deployment with payment processing must have **SSL Certificate** to secure financial transactions.

Command Line Environment

- Windows PowerShell 5.1+ or PowerShell Core 7+ **for command execution.**

2. The Hardware Requirements

Development Workstation

Local development requires minimum specification including:

- **Memory Limit:** 512MB minimum (1GB recommended) - configure in php.ini
- **Execution Time:** 300 seconds for large file uploads and migrations - set in php.ini
- **Storage:** 10GB minimum for development, scalable cloud storage for production.

Production Environment

- **Scalable storage** (cloud-managed storage recommended).
- **Caching system:** Redis for Windows recommended; file-based cache acceptable for non-production environments.

System Design

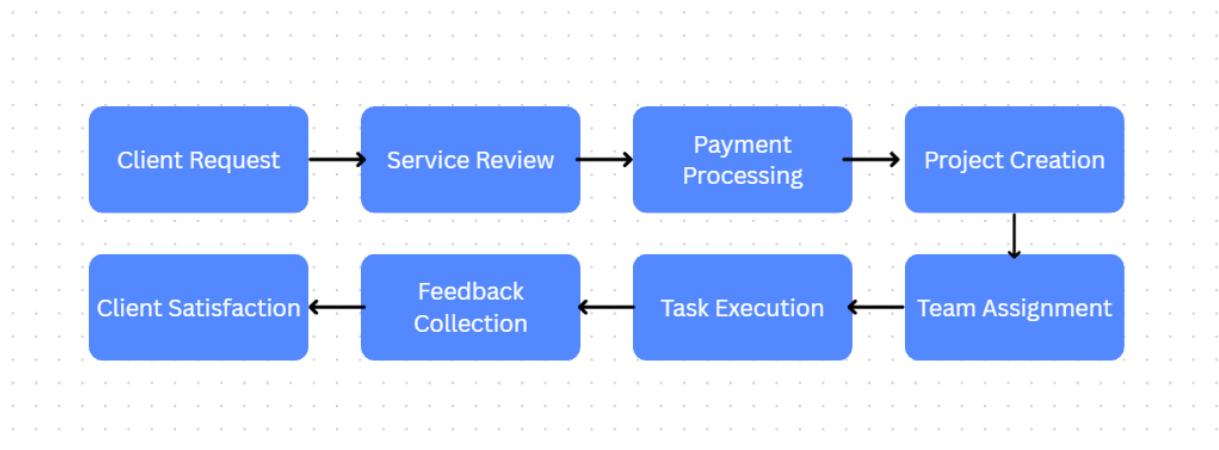


Figure 1. System Design

The figure 1 shows the system design that TAS uses to visually demonstrate the flow of project management from creating service requests of the client-to-client satisfaction. The system design tends to show the main courses of the system by breaking it down into seven categories

Detailed Process Flow

Phase 1: Service Request Initiation

Client Submission can be created through public form or in client portal. Once the system receives the information, it compiles all necessary requirements, including the specification of the service and any attached files uploaded by the client. The system will perform initial assessment by automatically categorizing the project or request and assigning its priority level. After that, the administrator will be notified and the request form is well organized into the queue for proper tracking and handling.

Phase 2: Administrative Review & Approval

The system includes an expert review point, an area of specialists that analyze the requirements. After that, the budget will be estimated precisely based on the project scope and complexity. The request then flows and is examined for approval, where it may be approved or rejected with clear explanations. Clients will receive automated updates throughout the process about the status of their request and the next steps they need to take.

Phase 3: Payment Processing & Verification

The system offers secure payment requests through Maya gateway, allowing clients to have safe transactions. Once a payment is made, the system executes a real-time verification using webhooks to confirm validity. The transactions executed by the system are automatically recorded to maintain the clarity of the financial audit trail and to ensure transparency with the clients. After the payment is confirmed, the workflow flows automatically to the next step in the process.

Phase 4: Project Creation & Team Assembly

Once a project is paid, the system declares it as an active project. It then selects the best and suitable adiutors based on their skills and availability. The system also allocates the project budget and plans the necessary resources. Finally, it sets up the project phases and defines key milestones to guide the execution of the work.

Phase 5: Task Breakdown & Assignment

The system breaks down each project into manageable tasks to ensure efficient execution. It assigns these tasks to team members and balances the workloads. Budgets are distributed for each task depending on its level, to ensure proper financial tracking. And lastly, the system establishes progress and defines success criteria to guide it.

Phase 6: Execution & Monitoring

To make sure that the status updates and progress are visible the system provides a real-time tracking. Reports go through regular reviews and quality checking to maintain the high standard of a project. The system offers structured channels to ensure that the client and the project team are aligned with the same goal. When issues arise, the system detects the problem and guides the team through effective resolutions.

Phase 7: Completion & Satisfaction

To ensure the quality of a project the system conducts a complete review before sending it to the client and asking for approval. Satisfaction surveys collect detailed feedback to identify the areas that need improvements. Throughout the process, important ideas and documentation are recorded to support future projects as well. Lastly, the system helps to maintain the strong bond between the clients and the project team members, which can lead to opportunities for future collaboration.

Entity Relationship Diagram

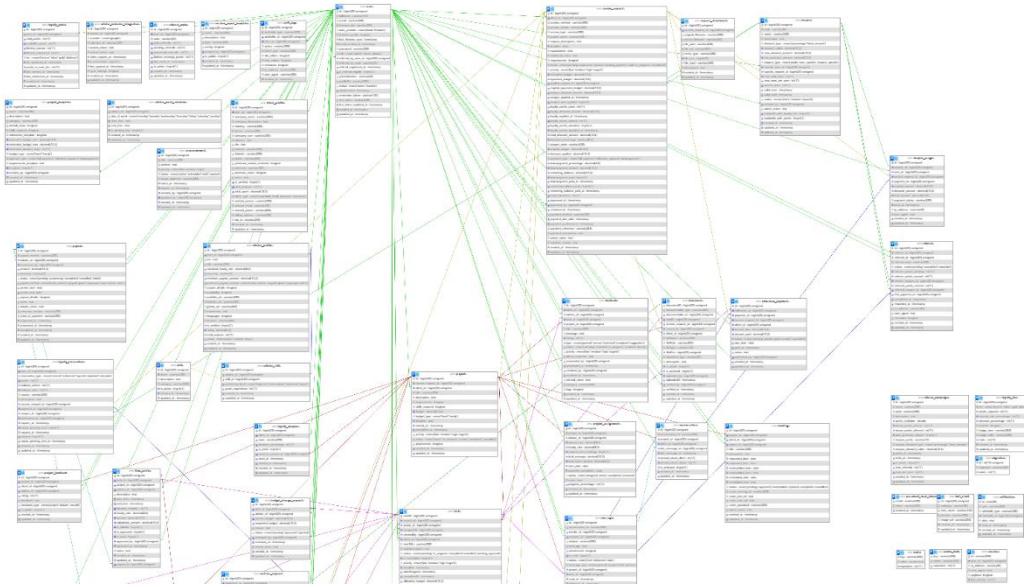


Figure 2. Entity-Relationship Diagram (1)

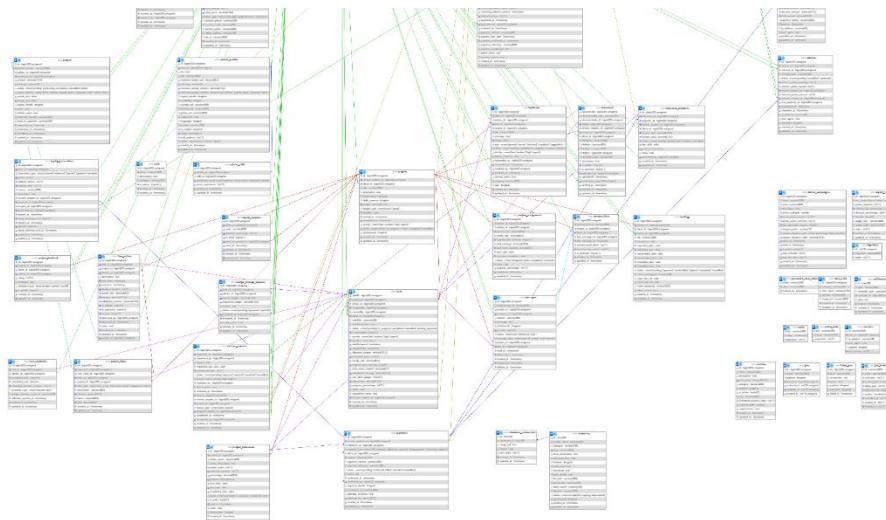


Figure 3. Entity-Relationship Diagram (2)

Use Case Diagram

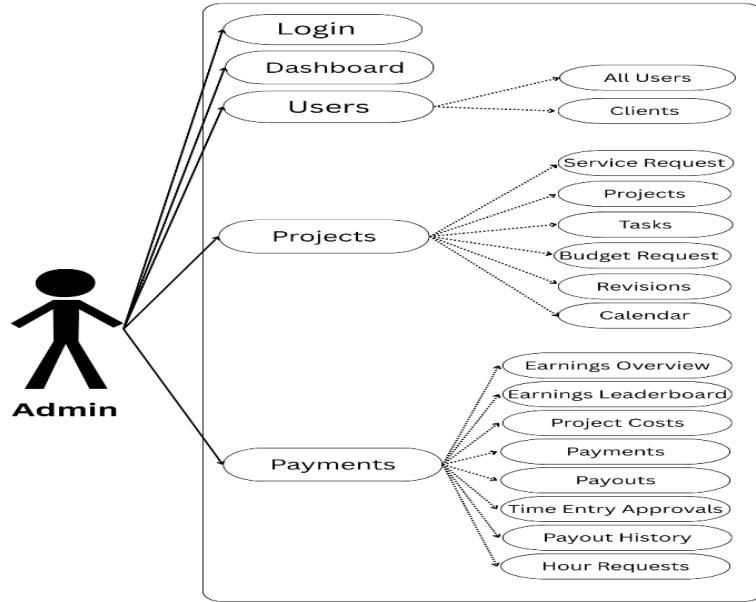


Figure 4. Use Case Diagram for the Admin (1)

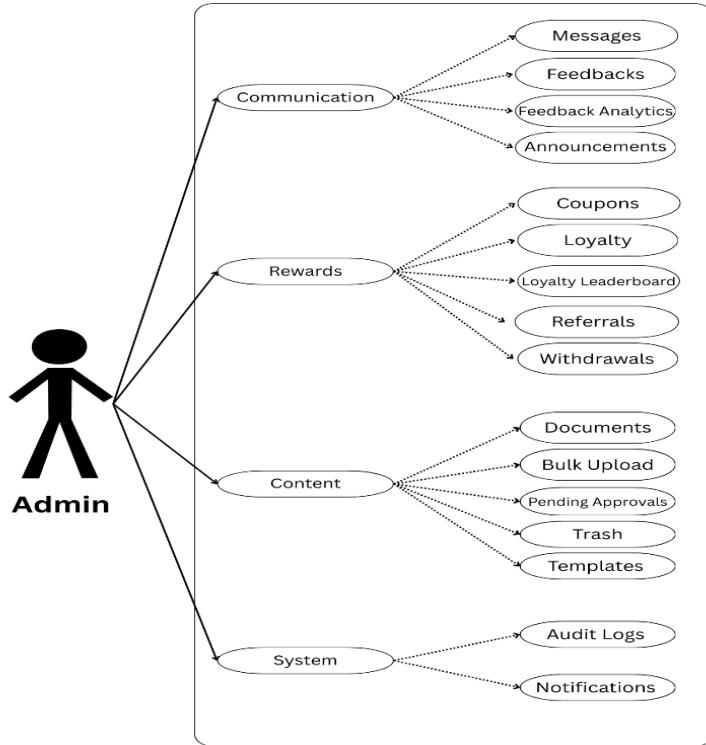


Figure 5. Use Case Diagram for the Admin (2)

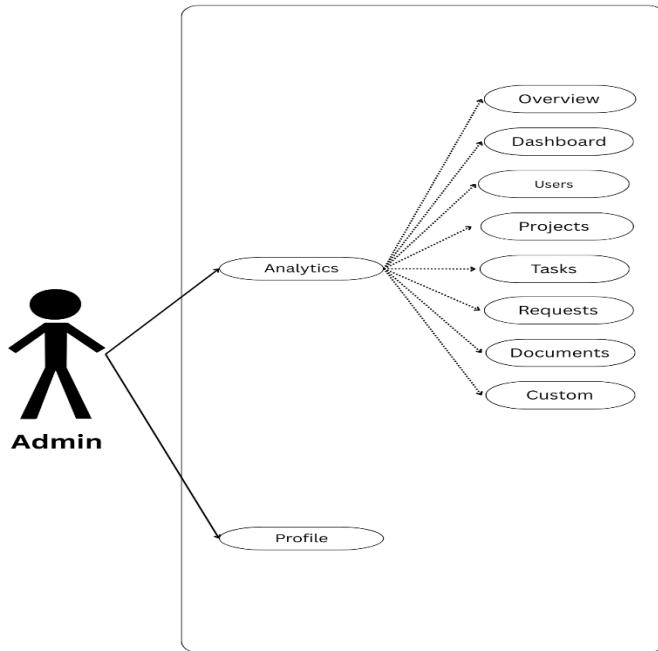


Figure 6. Use Case Diagram for the Admin (3)

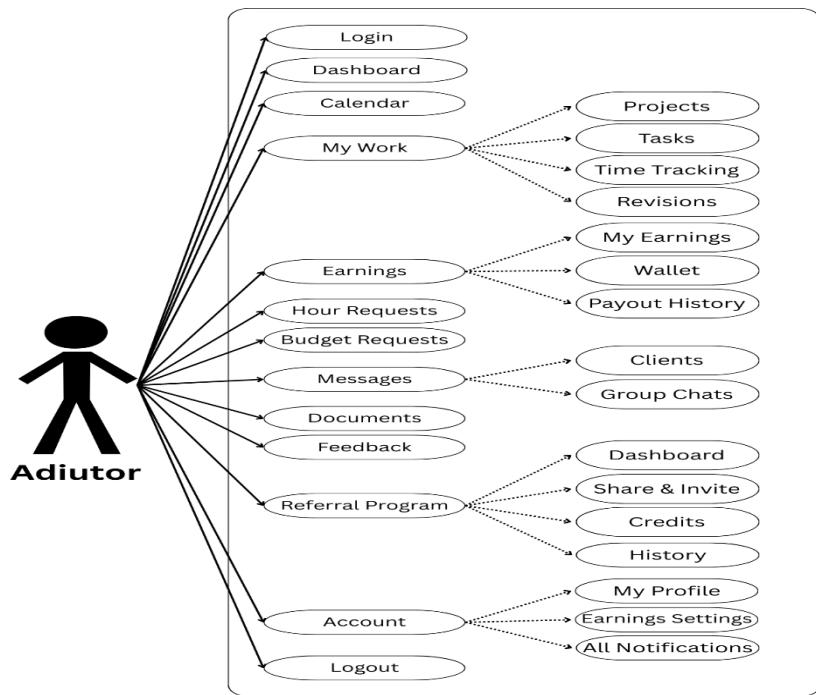


Figure 7. Use Case Diagram for the Adiutor

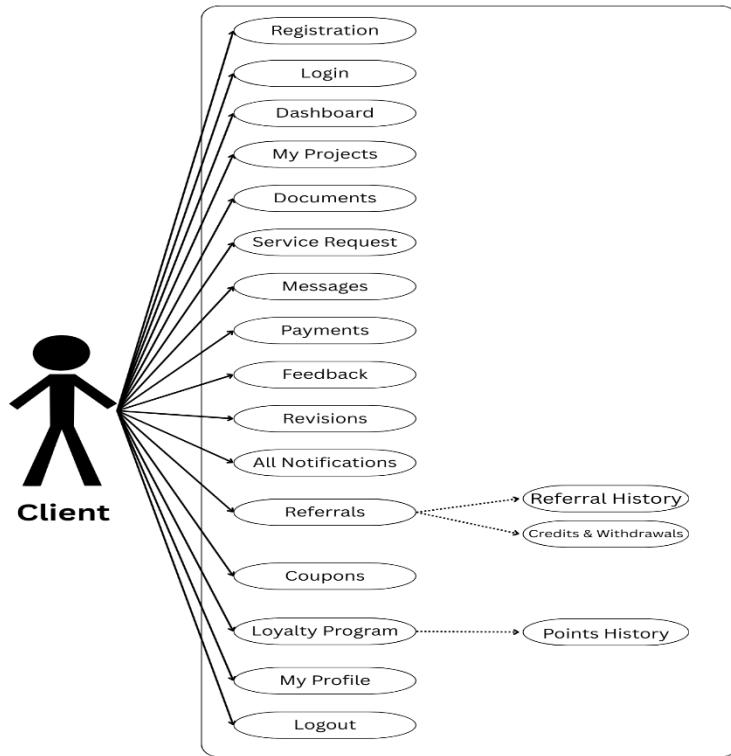


Figure 8. Use Case Diagram for the Client

API Integration

Treis Adiutor System (TAS) integrates different external services enabling authentication, payment processing, scheduling and online meetings, real-time messaging, and AI assistance. These APIs ensure confidentiality and integrity of transmitted data. Each integration contributes to automation, improved operational efficiency and provides seamless communication across system users.

Authentication and Security

Firebase Authentication API

The purpose of this API is to provide a secure authentication mechanism including Single Sign On (SSO). This integrates in the system by sending authentication requests to the firebase to validate user identity and create access tokens for role-based authorization.

Endpoints	Request Type	Purpose
/auth/login	POST	Authenticates credentials.
/auth/callback	GET	Handles Oauth social login redirects.
/auth/logout	POST	Session Termination
/auth/user	GET	User Profile retrieval

The data exchanged consists of user credentials, authentication tokens, profile information, and assigned access roles. The method of integration involves token authorization, where the frontend receives a secure Firebase ID Token that will be verified by the backend before granting system access.

Google reCAPTCHA API

The Google reCAPTCHA API is used to verify whether a form submission is performed by a human. A reCAPTCHA token is generated through the client-side widget and is then validated by sending a server-side request to Google's

verification service. The system transmits the user's response token along with server credentials to confirm that the request is not automated.

Payment Processing

Maya Business API

Endpoints	Request Type	Purpose
/payment/create	POST	Generates the payment link.
/payment/status	GET	Check transaction state.
/payment/webhook	POST	Sends real-time payment status updates

The purpose of this is to handle secure payment transactions for service orders and milestone billing. The data exchanged consists of payment amount, currency, payer details, transaction data, and status codes. Additionally, all transactions are protected and secure through PCI-DSS, Secure tokenization, and HTTPS.

Artificial Intelligence and Machine Learning

Google Gemini AI API

The Google Gemini AI API provides intelligent chatbot features for support and guided navigation, and FAQ responses. This integrates the system by sending a user prompt to Gemini AI and receives auto generated responses which are displayed in a chat widget. The data exchanged contains user message text, AI generated reply, and context history.

```
/**
 * Perform AI-powered service search using Gemini API
 */
public function search(Request $request)
{
    $request->validate([
        'query' => 'required|string|max:500',
        'services' => 'required|array'
    ]);

    $query = $request->input('query');
    $services = $request->input('services');

    // Get Gemini API configuration
    $geminiApiKey = config('services.gemini.api_key');
    $geminiModel = config('services.gemini.model', 'gemini-2.0-flash');
    $geminiApiUrl = config('services.gemini.api_url');

    if (!$geminiApiKey) {
        return response()->json([
            'error' => 'Gemini API key not configured'
        ], 500);
    }
}
```

Figure 9. Code Snippet for the Google Gemini AI API

Email and Communication API

Brevo SMTP API

This API handles transactional and bulk email notifications including registration confirmations, project updates, and payment receipts. This integrates

TAS issues POST request to Brevo's SMTP endpoint using API Key authentication. The service returns a delivery status and tracking information such as open and click rates.

```
// Send email notification if task is assigned to someone
if ($request->assignedTo) {
    try {
        $assignee = User::findOrFail($request->assignedTo);
        $assignedBy = Auth::user();
        Mail::to($assignee->email)->send(new TaskAssigned($task, $assignee, $assignedBy));
    } catch (\Exception $e) {
        // Log the error but don't fail the task creation
        \Log::error('Failed to send task assignment email: ' . $e->getMessage());
    }
}
```

Figure 10. Code Snippet for the Brevo SMTP API

Firebase Cloud Messaging (FCM)

The purpose of this API is to provide instant push notifications for project updates, task assignments, and payment confirmations. The system transmits a POST request containing a notification payload and the recipient's device or browser token. The notification is delivered instantly to the intended user, supporting role-based targeting across clients, Adiutors, and administrators.

The data exchanged includes the user's device or browser notification token, the notification title and body, routing or click-action data, and optional custom payload values used to personalize the message according to the user role. The integration is performed by generating secure server-to-server POST requests from the backend using a valid FCM server key, after which the message is

transmitted to Firebase for dispatch to the intended recipients through web push notifications and future support for mobile app delivery.

Media & Storage Management

Cloudflare R2 API

The purpose of this API is to provide secure, scalable, and cost-efficient cloud storage for project documents, user profile images, attachments, and other media resources, enabling global content delivery through a built-in content distribution network (CDN). The system issues S3-compatible file operations by sending upload requests to the R2 object storage bucket endpoint, retrieval requests to download stored files, and deletion requests to remove outdated or replaced content.

The data exchanged includes binary file contents, file metadata, directory or object paths, access credentials, and optional version control information for maintaining document history.

The integration is achieved by utilizing secure, signed HTTP requests that conform to the S3 REST API specification, allowing the platform to seamlessly upload, retrieve, and manage stored files with automated backup and edge caching for enhanced performance.

```

/**
 * Upload a file to Cloudflare R2
 *
 * @param UploadedFile $file
 * @param string $directory
 * @param string|null $filename
 * @param array $metadata
 * @return array
 */
public function uploadFile(UploadedFile $file, string $directory, ?string $filename = null, array $metadata = []): array
{
    try {
        // Generate filename if not provided
        if (!$filename) {
            $filename = $this->generateUniqueFileName($file);
        }

        // Ensure directory ends with /
        $directory = rtrim($directory, '/') . '/';

        // Full path in bucket
        $path = $directory . $filename;

        // Upload to R2
        $result = $this->disk->put($path, file_get_contents($file->getRealPath()), [
            'visibility' => 'public',
            'contentType' => $file->getMimeType(),
            'Metadata' => array_merge([
                'original_name' => $file->getClientOriginalName(),
                'upload_timestamp' => now()->toISOString(),
            ],
            $metadata
        ]);
    }
}

```

Figure 11. Code Snippet for the Cloudflare R2 API (1)

```

        'visibility' => 'public',
        'ContentType' => $file->getMimeType(),
        'Metadata' => array_merge([
            'original_name' => $file->getClientOriginalName(),
            'upload_timestamp' => now()->toISOString(),
        ], $metadata)
    ]);

    if (!$result) {
        throw new Exception('Failed to upload file to R2');
    }

    return [
        'success' => true,
        'path' => $path,
        'url' => $this->getPublicUrl($path),
        'size' => $file->getSize(),
        'mime_type' => $file->getMimeType(),
        'original_name' => $file->getClientOriginalName(),
        'stored_name' => $filename,
    ];
}

} catch (Exception $e) {
    Log::error('R2 Upload Error: ' . $e->getMessage(), [
        'file' => $file->getClientOriginalName(),
        'directory' => $directory,
        'filename' => $filename,
    ]);
}

```

Figure 12. Code Snippet for the Cloudflare R2 API (2)

Calendar and Scheduling

Google Calendar API

The Google Calendar API enables automated scheduling of project deadlines, milestone tracking, and availability management, ensuring that all parties involved in a project are notified and reminded of important dates and meetings. The system creates calendar events by sending a POST request to the Google Calendar events endpoint, retrieves calendar information through a GET request, and may update or delete events through corresponding PUT or DELETE service calls.

The data exchanged includes event titles, date and time range, participant email addresses, reminder configurations, availability details, and synchronization metadata used to maintain real-time schedule consistency. The integration is handled through OAuth-secured HTTP requests, allowing the system to automatically create deadlines, send reminders, synchronize events in real-time, and detect schedule conflicts across connected calendars.

Video Conferencing

Zoom Meeting API

The Zoom API supports the scheduling and management of virtual meetings and project consultation sessions, enabling clients and Adiutors to conduct remote reviews, discussions, and progress evaluations. The system creates online meetings by sending a POST request to the Zoom meeting creation

endpoint and retrieves meeting information by issuing a GET request to the corresponding meeting details endpoint.

The data exchanged includes meeting topic, duration, schedule timestamps, meeting URL, participant settings, host authentication data, and access control parameters such as waiting room configuration and passcodes. The integration is conducted through secure OAuth or JWT-authenticated API requests, enabling the system to automatically generate and deliver meeting links, synchronize sessions with project milestones, and support calendar integration for participant reminders.

Data Flow and User Interface

The data flow and user interface of Project Execution and Management System illustrates how information and data processes across the platform.

It explains the journey of data—from the moment a user provides input, through API and server processing. It enables sharing the necessary data and communication back to the user. TAS follows client-server architecture where a network is structured to include multiple clients and a central server. Clients are devices or programs that make request services, while the server is a powerful machine that provides these services. This flow ensures efficient data management and provides user experience with different users.

The system integrates multiple APIs to communicate in terms of a request and response between a client and server. The API is the bridge establishing the connection between them.

Data Flow Overview

1.1 User Input

The users interact with the system through web-based interfaces. Inputs include:

- Login credentials
- Service request details
- File and image uploads
- Payment Actions
- Messaging, Communication, and Feedback

1.2 API

After user interaction, the data flow moves through API to verify, secure, or enrich the data:

- **Firebase Authentication** – Verifies user's identity.
- **Google reCAPTCHA** – Validates human interaction.
- **Maya API** – Handles payments.
- **Gemini AI** – Processes chatbot queries.
- **Cloudflare R2** – Manages file uploads.
- **Google Calendar** – Schedules and syncs events.
- **Zoom API** – Creates online meeting sessions

1.3 Backend Layer

The backend layer, built using Laravel and PHP, is responsible for handling the system's core business logic. It performs data validation and sanitization to

ensure accuracy and security before processing any user request. The backend also manages data storage and retrieval operations through MySQL, enabling reliable and efficient database interactions. Additionally, it triggers real-time notifications via Firebase Cloud Messaging and generates dynamic dashboard updates, ensuring users receive timely information and an up-to-date system experience.

1.4 Server Response

The system then sends a response back to the client, allowing the front-end to render the updated information. This includes status confirmations for actions performed, progress updates related to ongoing projects, current payment status, and any messages or alerts that the user needs to see. Additionally, the system returns data used to update dashboard visualizations, ensuring the client always views accurate and real-time information.

Typical User Journey Through TAS

2.1 Client Journey

- Creating a service request
- Monitors project progress
- Select payments
- Submits comments, ratings and reviews

2.2 Adiutor Journey

- Viewing assigned projects
- Updates task progress
- Uploads documents

2.3 Administrator Journey

- Monitors all projects and tasks
- Assigning a project to adiutor
- Oversees the payment
- Generates system analytics
- Manages all user roles and platform settings

Data Flow Diagram

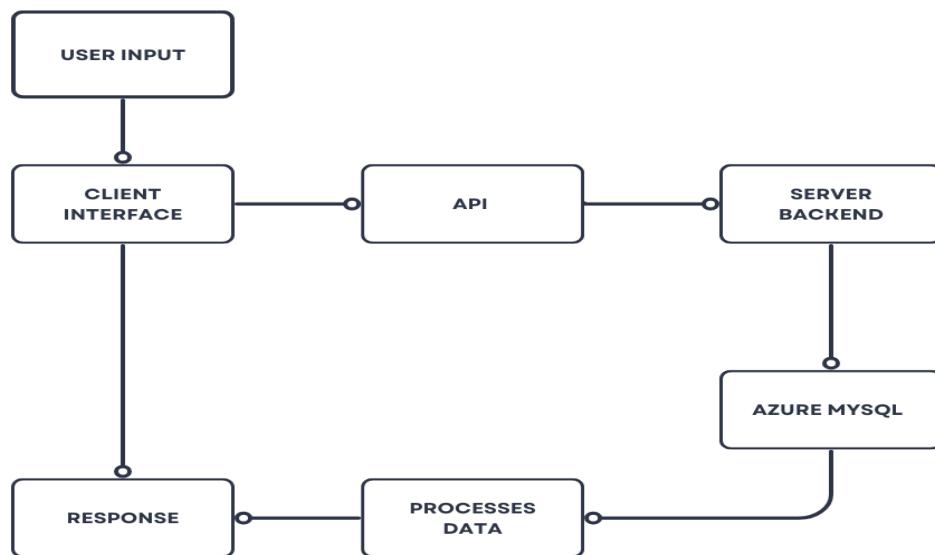


Figure 13. Data Flow Diagram

The diagram illustrates how data moves through the TAS, starting from the user input and ending with the system response. The data flow begins when the user enters information into the system, such as logging in, submitting a project request, uploading files, or performing other actions. This input is first handled by

the client interface, which validates the data and triggers the necessary API calls. The request is then sent through the API layer, which securely transmits the information from the client to the server backend. Once received, the server backend processes the request by executing business logic, interacting with external services, and communicating with the MySQL database whenever data needs to be stored, updated, or retrieved. After the database returns the required information, the server continues processing and formats the data into a structured response. This processed output is then returned to the client as a response, allowing the interface to update in real time by showing confirmations, progress updates, notifications, and other relevant information.

Project Processing Data Flow

The screenshot shows the 'Get Started with Your Project' form on the Treis Adiutor website. The form is divided into several sections:

- Contact Information:** Fields for Full Name (John Doe), Email Address (john@example.com), Phone Number (+1 (555) 000-0000), and Preferred Contact Method (Select a method: Email, Phone, or Messenger Link).
- Project Details:** Fields for Project Name (e.g., E-commerce Website Redesign), Category (Select a service category: Choose the category that best matches your project needs), Description (Describe your project: Tell us about your project goals, requirements, timeline, and any specific needs...), Deadline (Optional: dd/mm/yyyy), and Expectations (Optional: What are your expectations for this project?).
- Additional Notes:** A field for Any other information? (Optional: Any additional comments, questions, or special requirements...).
- Attachments (Optional):** A section for uploading relevant files, with a note that documents, images, or any relevant files (max 10MB each) can be uploaded.
- Security Verification:** A CAPTCHA field labeled 'I'm not a robot'.
- Buttons:** A 'Submit Request' button at the bottom of the form.
- Sidebar (How It Works):**
 - Submit Your Request: Tell us about your project needs
 - Get Matched: Find the perfect Adiutor for you
 - Start Collaborating: Work together to bring your project to life
- Sidebar (What to Expect):**
 - Response within 24 hours
 - Free initial consultation
 - Secure & confidential
 - Flexible payment options
- Sidebar (Need Help):**

Need assistance before submitting? We're here to help. [Contact Us →](#)

Figure 14. Public Client Creating a request

The figure 14 shows the form for creating a service request that is to be submitted to the admin for approval.

Client Requests

Total Requests: 65 | Pending: 21 | Approved: 12 | Rejected: 1

Search & Filters

Search	Status	Priority	Type	Client
Search requests...	Select an option	Select an option	Select an option	Select an option

Client Requests

REQUEST	CLIENT	TYPE	DEADLINE	STATUS	PRIORITY	DATE	ACTIONS
Mobile App Development	Mark Andrew Soliman	Programming	Not set	Paid	Medium	Dec 10, 2025	⋮
Custom Website Development	Paloma Turner	Programming	Not set	Completed	Medium	Dec 10, 2025	⋮
FinAlign - AI-Powered Personal...	Paloma Turner	Programming	May 30, 2026 (0 days)	In Progress	Medium	Dec 09, 2025	⋮
Website Redesign	John Arthur Lopez	Design	Not set	Paid	Medium	Dec 09, 2025	⋮
Mobile App Development	John Arthur Lopez	Programming	Not set	Approved	Medium	Dec 09, 2025	⋮
Custom Website Development	John Arthur Lopez	Programming	Not set	Paid	Medium	Dec 09, 2025	⋮
SEO Optimization	John Arthur Lopez	Marketing	Not set	Paid	Medium	Dec 09, 2025	⋮
Website Redesign	John Arthur Lopez	Design	Not set	Paid	Medium	Dec 09, 2025	⋮
SEO Optimization	Mark Andrew Soliman	Marketing	Not set	Paid	Medium	Dec 09, 2025	⋮
Project Execution and Management...	Mark Andrew Soliman	Programming	Not set	Paid	High	Dec 09, 2025	⋮
Website Redesign	John Arthur Lopez	Design	Not set	Paid	Medium	Dec 09, 2025	⋮
CloudVault - Enterprise Data E...	Mark Andrew Soliman	Programming	Not set	Paid	Medium	Dec 08, 2025	⋮
Web Application Development	Mark Andrew Soliman	Programming	Not set	In Progress	High	Dec 08, 2025	⋮
Web Design	Admin Account	Programming	Dec 25, 2025 (0 days)	Pending	Medium	Dec 08, 2025	⋮ ✓ ✕
UI/UX Design	Mark Andrew Soliman	Design	Not set	Paid	Medium	Dec 08, 2025	⋮

Bulk Actions | Select Priority | Apply

Showing 1 to 15 of 65 results

Figure 15. Admin Receives the Request

Figure 15 shows the admin receives the service request and decides whether the request is approved or not based on its description and skills required. In this phase it shows that the request is visible to the admin's page.

The screenshot shows the 'Request Details' page for a project named 'CloudVault - Enterprise Data Encryption & Key Management Platform'. The project was submitted on December 08, 2025, and is categorized as 'Paid' with 'Medium Priority'. The request details include:

- Project Name:** CloudVault - Enterprise Data Encryption & Key Management Platform
- Project Description:** I'm interested in: Web Application Development. Custom web application development for business processes, dashboards, and data management.
- Additional details:**
- Service Requirements:**
 - WHAT'S INCLUDED:** Custom Functionality, User Authentication, Database Integration, API Development, Admin Panel, Reporting System, 4 Months Support.
 - SKILLS REQUIRED:** Full-Stack Development, Database Design, API Development, Security.
- Timeline:**
 - Request ID: 56
 - Submitted: Dec 08, 2025
 - Service Type: Programming
 - Contact Method: Email
 - Contact Details: markin@treisaditor.com
 - Status: Paid
 - Priority: Medium
 - Deadline: Not set
 - Reviewed: Dec 08, 2025
 - Approved by: Admin Account

Client Information: Mark Andrew Soliman (markin@treisaditor.com, Bronze Member, +639761022819, Client Since Dec 08, 2025). A link to 'View Client Profile' is provided.

Admin Notes: A text input field for adding notes, with a '+ Add Note' button. It currently displays 'No admin notes yet'.

Attached Files: No files attached.

Associated Tasks: A table showing tasks associated with the request. The table has columns: Task #, Status, Priority, Assigned To, Due Date. One task is listed: Task # 1, In progress, Low priority, Unassigned, Not set due date.

Figure 16. Admin Request Approval

Figure 16 shows the main view for approving a request. This is where the admin allocates the budget for the project.

The screenshot shows the 'My Projects' section of the Tres Auditor client interface. The page is titled 'My Projects' and features a grid of 15 project cards. Each card provides a summary of a project, including its name, status, budget, deadline, progress, and options to view details or leave feedback.

- Website Redesign**: Pending, \$250,000.00, Not set, 0%, Created Oct 8, 2025.
- Mobile App Development**: In Progress, \$100,000.00, Not set, 0%, Created Oct 8, 2025.
- Website Redesign**: Completed, \$100,000.00, Not set, 0%, Created Oct 8, 2025.
- Custom Website Development**: Completed, \$1,200,000.00, Not set, 0%, Created Oct 8, 2025.
- SEO Optimization**: Completed, \$1,000,000.00, Not set, 0%, Created Oct 8, 2025.
- Website Redesign**: Completed, \$1,000,000.00, Not set, 0%, Created Oct 8, 2025.
- SynthAI Autonomous Code Review and Technical Debt...**: Completed, \$400,000.00, Not set, 100%, Created Oct 8, 2025.
- Quandamind Zero Trust Security Orchestration Platform**: Completed, \$400,000.00, Not set, 0%, Created Oct 8, 2025.
- EdgeTeller Decentralized Content Delivery Network...**: Completed, \$250,000.00, Not set, 0%, Created Oct 8, 2025.
- NovaSync Vendor Intelligence Dashboard**: In Progress, \$400,000.00, Not set, 20%, Created Oct 8, 2025 - Started Oct 8, 2025.
- E-commerce 3**: Completed, \$400,000.00, Dec 4, 2025, 100%, Created Nov 14, 2025 - Started Nov 14, 2025.
- E-commerce 3**: Completed, \$400,000.00, Dec 6, 2025, 0%, Created Nov 18, 2025.
- Website SEO Optimization**: Completed, \$50,000.00, Dec 24, 2025, 0%, Created Nov 16, 2025 - Started Nov 16, 2025.
- E-commerce 2**: Completed, \$10,000.00, Not set, 0%, Created Nov 16, 2025 - Started Nov 16, 2025.
- AI Assistant Mobile App**: Completed, \$10,000.00, Mar 24, 2026, 0%, Created Nov 16, 2025.
- TechStartup Corporate Website**: Completed, \$50,000.00, Dec 24, 2025, 0%, Created Nov 16, 2025.

Figure 17. Project List

Figure 17 shows that the project is also visible on the client side after being approved by the admin. The purpose of this is for the client to be updated with all the progress of the project.

The screenshot shows the 'Projects Management' section of the Tres Advisor platform. On the left, a sidebar navigation includes 'Dashboard', 'MANAGEMENT' (with 'Users', 'Projects', 'Tasks', 'Budget Requests', 'Revisions', 'Calendar'), 'FINANCIAL' (with 'Payments'), 'ENGAGEMENT' (with 'Communication', 'Rewards'), 'CONTENT' (with 'Content'), 'SYSTEM' (with 'System', 'Analytics'), and 'Profile'. The main area is titled 'Projects Management' with the subtitle 'Manage and track all client projects'. It features a summary bar with metrics: Total Projects (42), Active (7), In Progress (2), In Review (0), Completed (33), and Cancelled (0). Below is a detailed table of projects:

PROJECT	CLIENT	STATUS	PRIORITY	BUDGET	TEAM	DATES	ACTIONS
Mobile App Development	Mark Andrew Soliman mkndrewsoliman@tresaditor.com	Active	Medium	\$600,000.00 \$470,000.00 -\$130,000.00	No team	Created: Dec 10, 2025	Edit Delete
Custom Website Development	Paloma Turner tekex@inncommerce.shop	Completed	Medium	\$600,000.00 \$470,000.00 -\$130,000.00		Created: Dec 10, 2025	Edit Delete
FinAlign - AI-Powered Personal Finance & Investment Platform (4+ Phases)	Paloma Turner tekex@inncommerce.shop	Active	Medium	\$600,000.00 \$425,000.00 -\$175,000.00		Created: Dec 09, 2025 Deadline: May 30, 2026	Edit Delete
Website Redesign	John Arthur Lopez john.smith@tresaditor.com	Active	Medium	\$300,000.00 \$307,190.00 +\$7,190.00	No team	Created: Dec 09, 2025	Edit Delete
Mobile App Development	John Arthur Lopez john.smith@tresaditor.com	Active	Medium	\$500,000.00	No team	Created: Dec 09, 2025	Edit Delete
Website Redesign	John Arthur Lopez john.smith@tresaditor.com	Completed	Medium	\$1,000,000.00	No team	Created: Dec 09, 2025	Edit Delete
Custom Website Development	John Arthur Lopez john.smith@tresaditor.com	Completed	Medium	\$1,000,000.00	No team	Created: Dec 09, 2025	Edit Delete
SEO Optimization	John Arthur Lopez john.smith@tresaditor.com	Completed	Medium	\$1,000,000.00	No team	Created: Dec 09, 2025	Edit Delete
Website Redesign	John Arthur Lopez john.smith@tresaditor.com	Completed	Medium	\$1,000,000.00	No team	Created: Dec 09, 2025	Edit Delete
SEO Optimization	Mark Andrew Soliman mkndrewsoliman@tresaditor.com	Active	Medium	\$127,039.30 \$72,960.70	No team	Created: Dec 09, 2025	Edit Delete
Project Execution and Management System	Mark Andrew Soliman mkndrewsoliman@tresaditor.com	Completed	High	\$450,000.00 \$425,000.00 -\$25,000.00	No team	Created: Dec 09, 2025	Edit Delete
CloudVault - Enterprise Data Encryption & Key Management Platform	Mark Andrew Soliman markiii@tresaditor.com	Active	Medium	\$500,000.00 \$425,000.00 -\$75,000.00		Created: Dec 08, 2025	Edit Delete
Web Application Development	Mark Andrew Soliman markiii@tresaditor.com	Active	High	\$1,000,000.00 \$971,520.00		Created: Dec 08, 2025	Edit Delete
UI/UX Design	Mark Andrew Soliman markiii@tresaditor.com	Completed	Medium	\$200,000.00	No team	Created: Dec 08, 2025	Edit Delete
API Development	Mark Andrew Soliman markiii@tresaditor.com	Completed	Medium	\$150,000.00 \$50,000.00	No team	Created: Dec 08, 2025	Edit Delete

Showing 1 to 15 of 42 results

Figure 18. Project Status

Figure 18 shows that the project approved in figure 3 is now visible to the project management page where all the projects are stored and managed by the administrator. In this page the admin is allowed to edit or delete a project.

The screenshot shows the 'Tasks Management' section of the Treis Adiutor application. The left sidebar contains a navigation menu with categories like Dashboard, MANAGEMENT, FINANCIAL, ENGAGEMENT, CONTENT, SYSTEM, and Profile. Under MANAGEMENT, 'Projects' is selected, which is further expanded to show 'Service Requests', 'Projects', 'Tasks', 'Budget Requests', 'Revisions', and 'Calendar'. The main content area is titled 'Task Management' and includes a sub-header 'Manage and track all tasks'. At the top right, there are buttons for '+ Create Task' and 'Bulk Actions', along with a user profile icon for 'Admin Account'. Below this, there are four summary boxes: 'Total Tasks 42' (blue), 'Pending Tasks 3' (orange), 'In Progress 8' (green), and 'Completed 31' (grey). A search bar and filter options are available above the task list. The task list itself has columns: TASK, CLIENT, ADIUTOR, STATUS, PRIORITY, DEADLINE, and ACTIONS. Each task row includes a checkbox, the task name, the client's name, the adiutor's name and profile picture, the current status (e.g., In Progress, Pending, Completed), priority level (Low, Medium, High), deadline (if set), and a set of actions buttons (edit, delete, etc.). The tasks listed include various projects such as 'Insurance Planning & Analysis', 'Multi-Currency & International Support', 'AI-Powered Financial Recommendations', and 'Project Infrastructure & Development Environment'. The interface uses a clean, modern design with a light blue and white color scheme.

Figure 19. Admin Creating Task

Figure 19 is about creating tasks and assigning adiutor for the project. In this phase we are able to create tasks in an existing project or delete it.

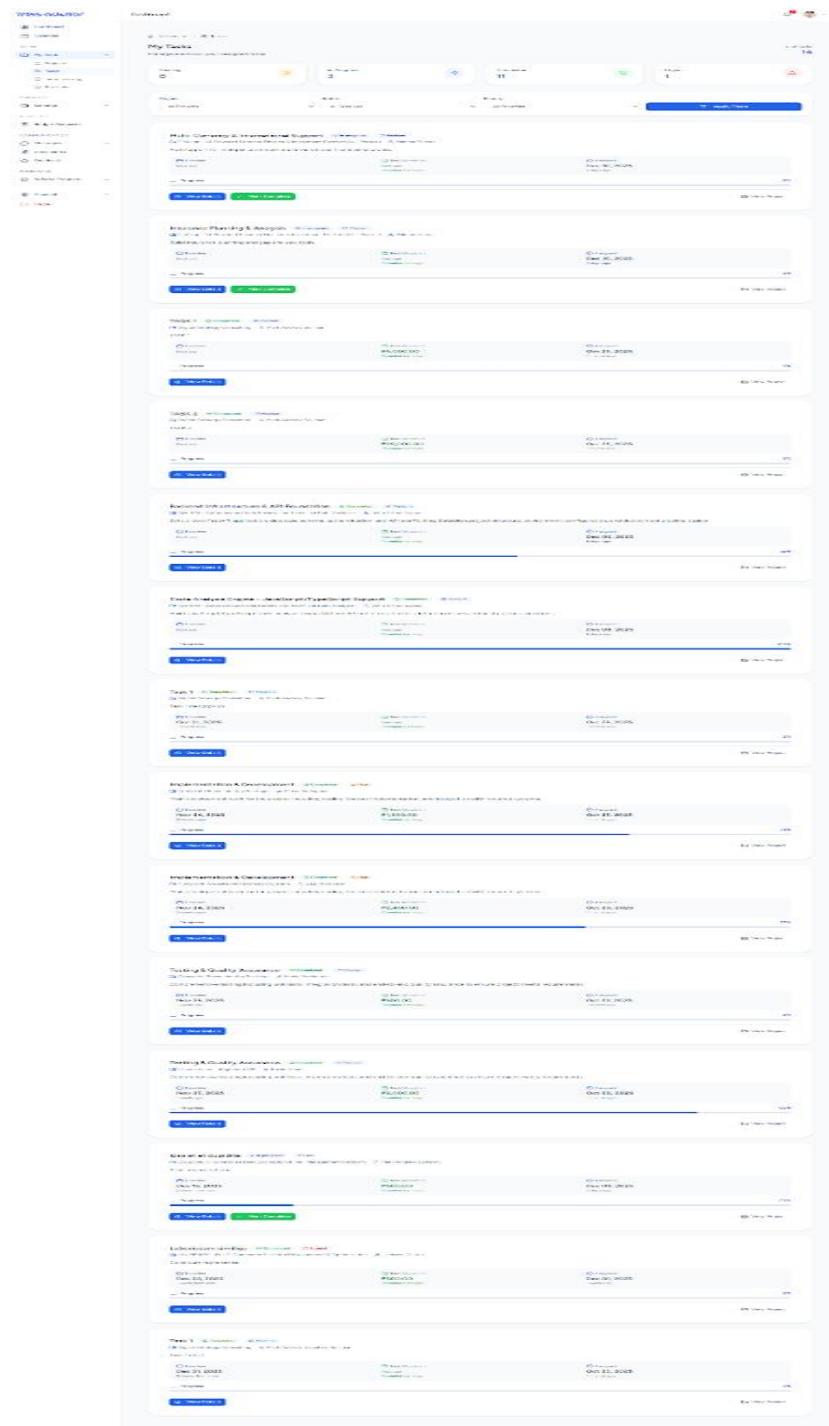


Figure 20. Adiutor Task Lists

In figure 20 the task is visible to the adiutor's side so he/she can start the creation of the task and earn depending on his/her rate.

CHAPTER III

IMPLEMENTATION

Technology Stack

Core Programming Languages & Frameworks

- **PHP 8.2.12** – Backend programming language
- **Laravel 12.33.0** – PHP web framework for backend development
- **Tailwind CSS** – Frontend styling framework
- **Alpine.js** – Lightweight JavaScript framework for frontend interactivity
- **Chart.js** – JavaScript library for interactive data visualization

Databases

- **SQLite** – Development database
- **MySQL** – Production database
- Features: High availability, automated backups, scaling, encryption at rest, SSL connections.

Web Servers

- **Apache / Nginx** – Web server for production deployment
- **Windows Server** – Deployment environment (Windows 10/11 or Server 2019/2022)

Authentication & Security

- **Firebase Authentication API** – Single Sign-On (SSO), social logins, multi-provider authentication
- **reCAPTCHA API (Google)** – Spam protection for forms
- Security: Role-based access control, secure session management

Payment Processing

- **Maya Business API** – Payment gateway for credit/debit cards, digital wallets, webhooks
- Security: PCI DSS compliance, secure tokenization

AI & Machine Learning

- **Google Gemini AI API** – Intelligent chatbot, NLP, context-aware responses, learning capabilities

Email & Communication

- **Brevo SMTP API (formerly Sendinblue)** – Transactional and bulk emails, templates, notifications

Real-Time Communication

- **Firebase Cloud Messaging (FCM)** – Push notifications, real-time updates, in-app messaging.

Media & Storage

- **Cloudflare R2 API** – Cloud storage with CDN, scalable file storage, edge caching

Calendar & Scheduling

- **Google Calendar API** – Event/deadline tracking, team availability synchronization

Video Conferencing

- **Zoom API** – Meeting scheduling, management, participant access, recording integration

Other Tools & Utilities

- **Composer** – PHP dependency management
- **Node.js 18+ with npm** – Frontend asset compilation
- **Redis** – Caching (recommended for production)
- **PowerShell 5.1+ / PowerShell Core 7+** – Command execution for Windows environments

Frontend/UX Integration

- Floating chat widget for AI support (Google Gemini AI)
- Rich file attachment and document management (Cloudflare R2)
- Interactive dashboards with Chart.js

Deployment & Hosting Requirements

- SSL Certificate (Let's Encrypt or commercial) for secure transactions
- 512MB–1GB memory minimum for development, scalable cloud storage for production.
- Execution time configuration (php.ini) for large uploads/migrations.

System Modules

System modules are the individual, organized parts of a software system that each perform a specific function. Instead of building one large, complicated system, it is divided into smaller modules—such as User Management, Payment Processing, Project Management, or Notifications—so each module focuses on one responsibility.

1. User Management Module

The User Management Module is responsible for all processes related to authentication, authorization, and user profiling. It uses Firebase Authentication for email-password, multi-provider SSO, and social logins. This module offers security for users while also managing profile customization, account statuses, and reCAPTCHA validation for enhanced security.

2. Service Request Management Module

This module receives, performs, and processes the submission of requests from clients. It supports all kinds of requests that may come from a public user or

an authenticated client user, rich file attachments via Cloudflare R2, and sort the request based on priority. Administrators can review, approve, or reject requests, while clients receive real-time updates through Firebase. TAS also integrates budget estimation and approval of workflows to ensure transparency of request processing.

3. Payment Processing Module

The Payment Processing Module is responsible for all payment transactions using Maya Business API. It generates a secure payment link, digital wallet transactions, and real-time webhook verification for successful payments. This module offers a full audit trail of all transactions, ensures PCI-compliant processing, and enables synchronization of finance for project creation and budgeting.

4. Project Management Module

This module handles the entire project lifecycle from initiation to completion. When the administrator approved a project and confirms successful payment of the client it automatically generates it to a active project and choose a adiutor based on skills and availability. The system enables timeline planning, step-by-step tasks, milestone creation, budget monitoring, and real-time status progression. TAS integrated Google Calendar for proper scheduling and adherence to deadlines, while Firebase provides the consistency of project status updates.

5. Task Management Module

This module enables administrators and adiutors to break projects into small manageable tasks. It shows the availability of adiutors, assigns tasks smartly, and tracks the progress through percentage-based updates. This module allocates the budget for a certain task, edit budget changes, and structured completion process. Zoom meeting integration is included to help adiutors to demonstrate virtually the tasks, progress, or changes in a project for validation of the client.

6. AI Chat Support Module

This module simply provides automated responses for customer support; it is using Google Gemini AI for better responses. It powers a context-aware chatbot that delivers smart responses to common inquiries across all public pages. With one-hour memory caching, the system maintains conversation context for improved user experience. The floating widget integrates seamlessly with the website and utilizes the Gemini API for accurate, optimized responses.

7. Document & File Management Module

This module keeps all the files in a centralized storage and can be accessed within the system. Using Cloudflare R2 with global CDN optimization, it allows users to upload, download, and manage documents related to service requests, projects, and tasks. This module ensures secure, role-based access control,

metadata indexing, version history tracking, and automatic cloud-based backups to maintain file integrity.

8. Notification & Communication Module

This module handles all the notifications across all the users. It uses Firebase Cloud Messaging for real-time push alerts and Brevo SMTP for email notifications, ensuring that users will receive notifications, updates, deadlines, and system events.

9. Scheduling & Calendar Module

This module integrates Google Calendar for precise scheduling and deadline managing. It automates the creation of progress for milestones and tasks, shows the availability of an adiutor, and triggers reminders for upcoming deadlines. This module is responsible for synchronization of all project timelines, and that users follow a well-defined schedule.

10. Video Conferencing Module

This module uses the Zoom API to facilitate online meetings within the platform. It automatically generates meeting links, manages participants, and embeds schedules into the Google Calendar system. The module supports project reviews, task evaluations, and client-team discussions, providing users with integrated access to meeting recordings and session summaries.

11. Feedback & Rating Module

This module collects, manages, and displays the client feedback whether a project meets their expectation and enriches their satisfaction. It allows clients to submit ratings, write their opinions or feedback through comments, and participate in follow-up evaluations. Brevo handles automated feedback emails, while administrators can review responses, track satisfaction trends, and analyze past interactions to improve the quality of the service.

12. Financial Management Module

This module oversees all the operations about finance in the system. It manages project and task-level budget allocations, processes budget change requests, and integrates with the Maya Business API for real-time financial tracking. The module also includes revenue reporting, transaction histories, invoice generation, and automated financial workflows that support accurate financial monitoring and decision-making.

13. Time Tracking and Productivity Module

This module records the time spent on tasks using built-in timers or manual entries, supporting time-based billing calculations, productivity analytics, and detailed timesheet reports. Administrators can review and approve logged hours, enabling effective resource planning and performance assessment for both individuals and teams.

14. Reporting & Business Intelligence Module

This module provides advanced analytics and reporting tools to visualize system performance. Using Chart.js and MySQL Database insights, it generates dashboards, KPI metrics, and customizable reports. Users can export data in PDF, CSV, or Excel formats, view revenue trends, track task and project performance, and access real-time insights for strategic decision-making.

15. System Administration Module

The System Administration Module gives administrators full control over platform configurations and system-wide settings. It covers user management, project configuration, security policies, and notification controls. The module also maintains system activity logs, enforces access permissions, and monitors system performance to ensure continuous stability and compliance.

16. Cloud Infrastructure Management Module

This module manages backend infrastructure, including Cloudflare R2 for storage and Database for MySQL. It ensures scalable, secure, and optimized cloud operations. Responsibilities include database backups, SSL enforcement, resource scaling, caching with Redis, and monitoring system performance. It serves as the foundation that supports reliability, speed, and data integrity across the entire TAS ecosystem.

Testing and Debugging

Testing and debugging are essential phases in ensuring the reliability, functionality, and performance of the Treis Adiutor platform. The system utilizes a structured approach combining automated, integration, and manual testing to validate both individual components and the complete workflow. Debugging tools are employed to identify, analyze, and resolve issues efficiently.

1. Testing Approaches

1.1 Unit Testing

Unit Testing focuses on evaluating individual components in isolation to ensure that the core logic of each module operates as expected. This includes validating functions, data processing, and overall module-level behavior. A sample of this process can be seen in the *TimeEntryTest.php* file, which demonstrates how specific features are tested independently to confirm their correctness.

```

File Edit Selection View Go Run Terminal Help
tests > Unit > Models > TimeentryTest.php > VSC Intellisense > My TimeentryTest > test_formats_duration_correctly
1 You 5 days ago | author (You)
2
3 class TimeentryTest extends TestCase
4 {
5
6     /* Test that duration is calculated correctly */
7     public function test_calculates_duration_correctly(): void
8     {
9         $timeentry = Timeentry::factory()->create([
10             'start_time' => now(),
11             'end_time' => now(),
12             'duration_minutes' => 120,
13             'billable_minutes' => 120,
14             'rate' => 100,
15             'calculated_amount' => 1000,
16         ]);
17
18         $this->assertEquals(120, $timeentry->duration_minutes);
19         $this->assertEquals(1000, $timeentry->calculated_amount);
20     }
21
22     /* Test formatted duration output */
23     public function test_formats_duration_correctly(): void
24     {
25         $timeentry = Timeentry::factory()->create([
26             'start_time' => now(),
27             'end_time' => now(),
28             'duration_minutes' => 360, // 2h 30m
29             'rate' => 100,
30             'calculated_amount' => 3600,
31         ]);
32
33         $this->assertEquals('2h 30m', $timeentry->getFormattedDuration());
34
35         /* Test formatted duration for less than an hour */
36     }
37
38     public function test_formats_short_duration_correctly(): void
39     {
40         $timeentry = Timeentry::factory()->create([
41             'start_time' => now(),
42             'end_time' => now(),
43             'duration_minutes' => 10,
44             'rate' => 100,
45             'calculated_amount' => 100,
46         ]);
47
48         $this->assertEquals('10m', $timeentry->getFormattedDuration());
49     }
50 }

```

Figure 21. Screenshot for the Unit Testing

1.2 Feature/Integration Testing

Feature and Integration Testing involve examining complete HTTP request and response cycles to ensure that different modules interact correctly within real system workflows. This testing approach verifies end-to-end processes, confirming that all interconnected components function seamlessly when combined.

```
/*
 * Test that duration is calculated correctly
 */
public function test_calculates_duration_correctly(): void
{
    $timeEntry = TimeEntry::factory()->create([
        'start_time' => now()->subHours(2),
        'end_time' => now(),
        'duration_minutes' => 120,
        'billable_minutes' => 120,
        'hourly_rate' => 500,
        'calculated_amount' => 1000,
    ]);

    $this->assertEquals(120, $timeEntry->duration_minutes);
    $this->assertEquals(1000, $timeEntry->calculated_amount);
}
```

Figure 22. Screenshot for the Feature/Integration Testing (1)

```
public function test_admin_can_approve_time_entry(): void
{
    $timeEntry = $this->createTimeEntry([
        'duration_minutes' => 120,
        'hourly_rate' => 500,
        'calculated_amount' => 1000.00, // 2 hours * 500
    ]);

    $this->actingAs($this->admin);

    // Route: admin.payouts.approve-time-entry
    $response = $this->post(route('admin.payouts.approve-time-entry', ['id' => $timeEntry->id]));

    // Should redirect with success
    $response->assertStatus(302);

    // Verify time entry was approved
    $timeEntry->refresh();
    $this->assertTrue($timeEntry->is_approved);
    $this->assertEquals($this->admin->id, $timeEntry->approved_by);
    $this->assertNotNull($timeEntry->approved_at);
}
```

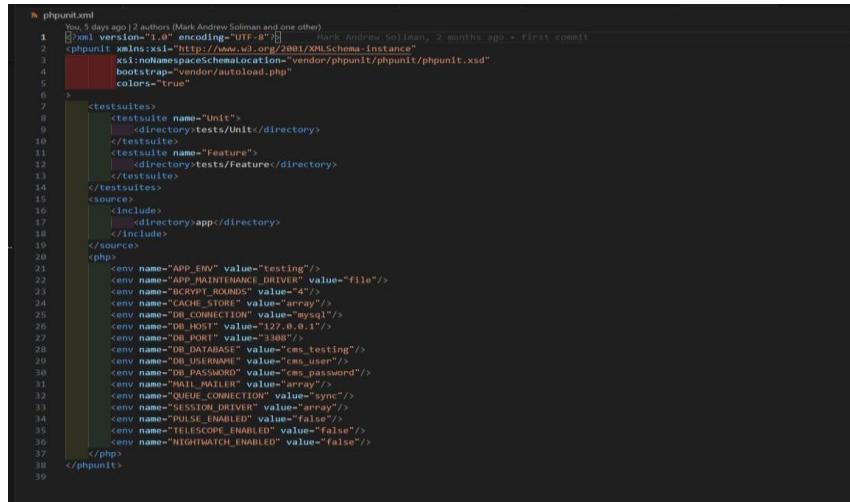
Figure 23. Screenshot for the Feature/Integration Testing (2)

1.3 Manual Testing

Manual Testing is performed to review visual elements and evaluate the system's overall UI/UX quality. This approach is particularly effective for assessing complex user interactions and conducting usability testing.

1.4 Testing Tools and Configuration

- Primary Testing Framework: PHPUnit



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:noNamespaceSchemaLocation="vendor/phpunit/phpunit/phpunit.xsd"
4      bootstrap="vendor/autoload.php"
5      colors="true"
6  >
7      <testsuites>
8          <testsuite name="Unit">
9              <directory>tests/Unit</directory>
10         </testsuite>
11         <testsuite name="Feature">
12             <directory>tests/Feature</directory>
13         </testsuite>
14     </testsuites>
15     <sources>
16         <include>
17             <directory>app</directory>
18         </include>
19     </sources>
20     <php>
21         <env name="APP_ENV" value="testing"/>
22         <env name="APP_DEBUG" value="true"/>
23         <env name="CACHE_STORE" value="array"/>
24         <env name="DB_CONNECTION" value="mysql"/>
25         <env name="DB_HOST" value="127.0.0.1"/>
26         <env name="DB_PORT" value="3306"/>
27         <env name="DATABASE" value="es_testing"/>
28         <env name="DB_USERNAME" value="es_user"/>
29         <env name="DB_PASSWORD" value="es_password"/>
30         <env name="MAIL_MAILER" value="array"/>
31         <env name="QUEUE_CONNECTION" value="sync"/>
32         <env name="QUEUE_DRIVER" value="array"/>
33         <env name="PULSE_ENABLED" value="false"/>
34         <env name="TELESCOPE_ENABLED" value="false"/>
35         <env name="NIGHTWATCH_ENABLED" value="false"/>
36     </php>
37 </phpunit>
38

```

Figure 24. Screenshot for the Testing Tools and Configuration

Challenges and Solutions

As for the challenges the developers had during the development process of the project, mainly it revolves around the debugging side for all of the features implemented in the project. Challenges also include the changing or adding of more tech stacks in some parts of the development process to make sure that the system is running smoothly and all features are functioning right.

Problem 1: Database Schema Mismatch

Issue: Laravel migrations were out of sync with actual production database schema due to localized development.

Solution: Created UseCmsSqlSchema trait that imports the actual cms.sql file for testing, ensuring tests run against the real database structure.

Problem 2: External API Dependencies

Issue: Tests requiring Maya payment gateway or Firebase notifications would make real API calls.

Solution: Created FakeMayaPaymentService and FakeFirebaseService mock classes that simulate external services with configurable success/failure states.

Problem 3: Controller Validation Order Bug

Issue: The PayoutManagementController::approveTimeEntry() method calls \$timeEntry->applyAdjustment() before validating that adjustment_reason is provided when adjust=true, causing a TypeError.

Problem 4: Parallel Test Execution

Issue: The Parallel Test Execution was slow; it takes minutes to execute the testing.

Solution: Added brianium/paratest package for parallel test execution to speed up the test suite.

Admin Flowchart

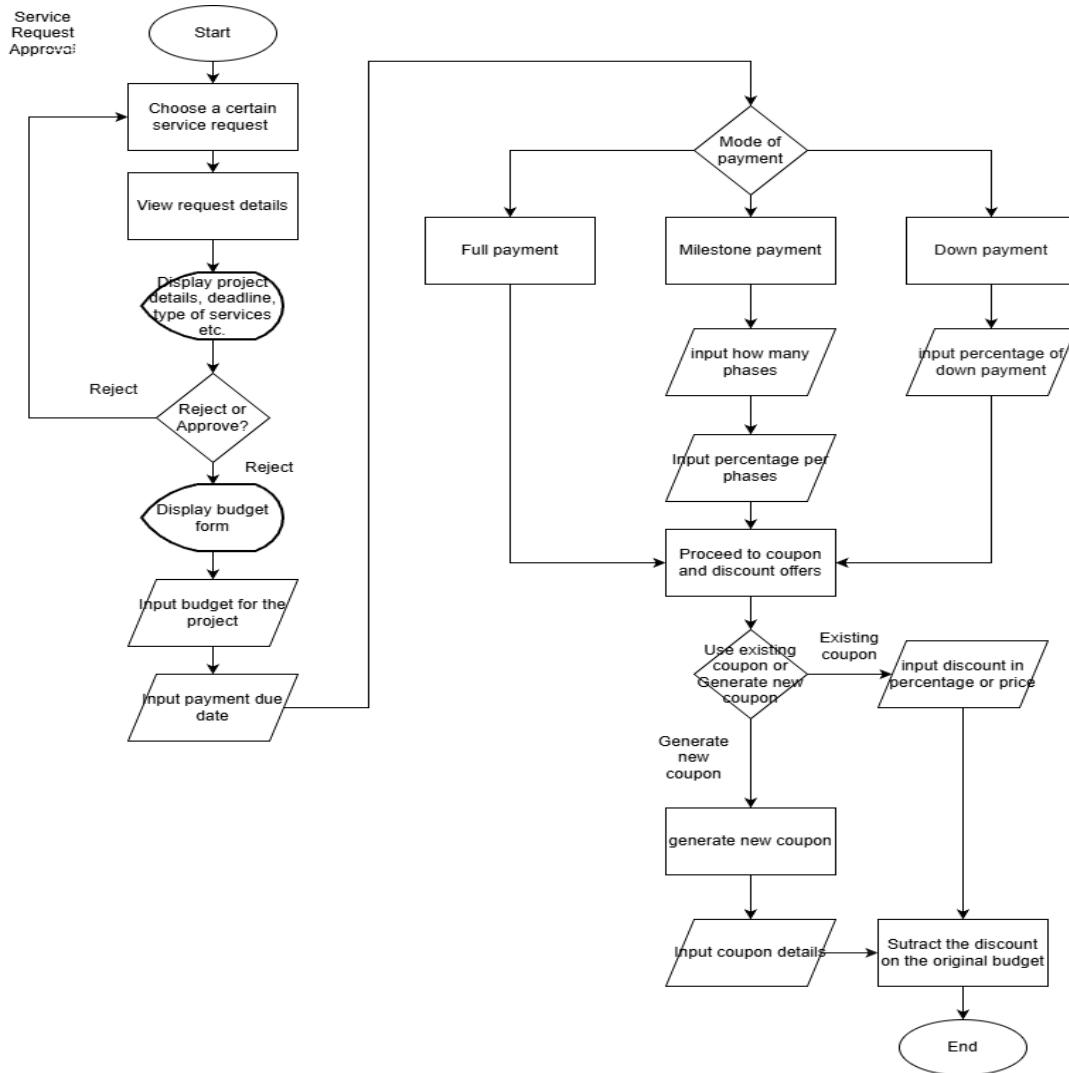


Figure 25. Project and Task Management

Figure 25 shows the process for managing a task within a specific project.

The admin selects and views a project. The admin can also create a sub task if required, assigned an adiutor, and sets a project budget.

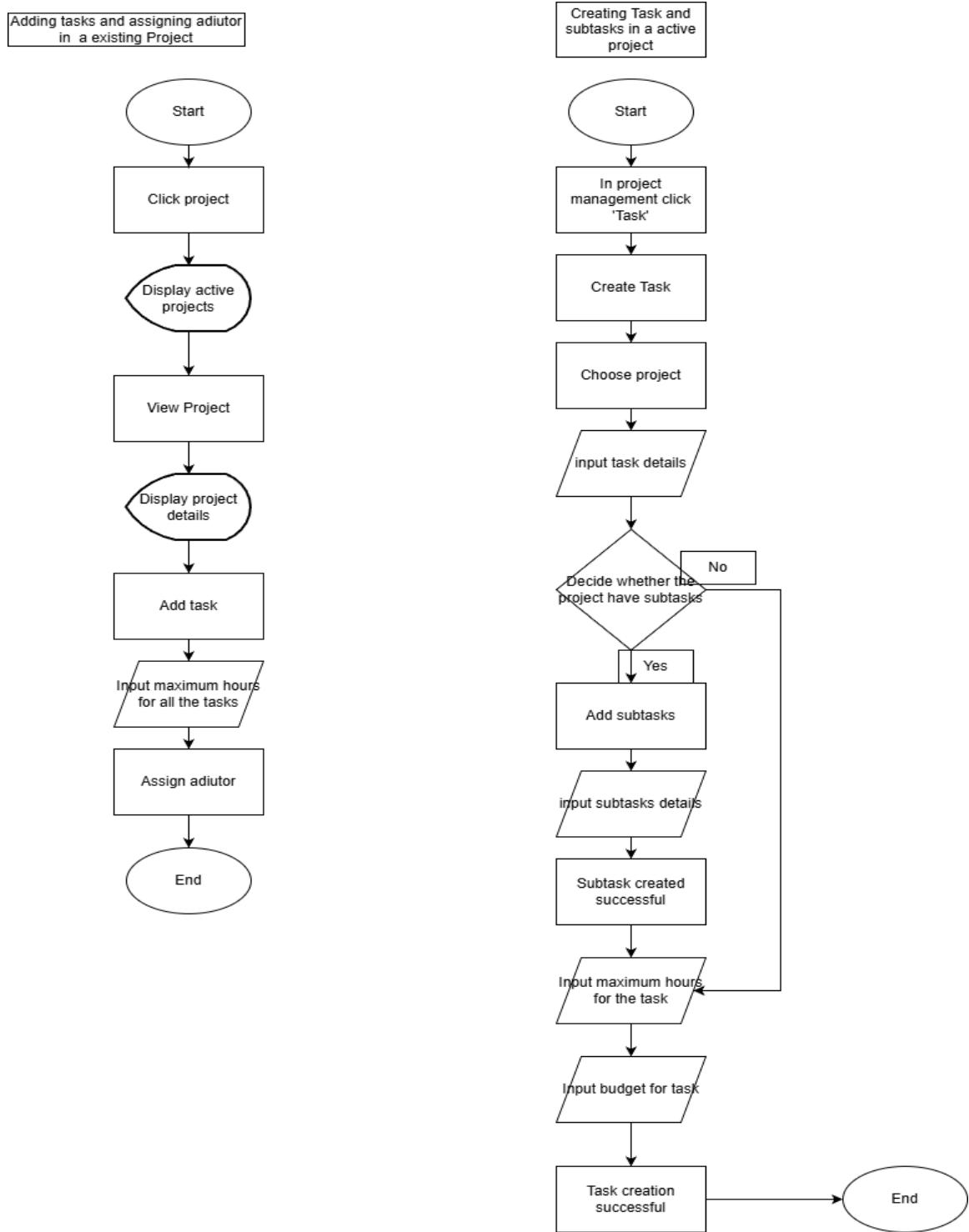


Figure 26. Service Request Approval

Figure 26 shows the service request approval made by the client. The diagram illustrates how the admin review and process service request. The admin can control whether to approve or reject it. Upon approval, the admin sets the budget and payment method, applies coupons or discounts, and finalizes the project budget.

Client Flowchart

Registration Page

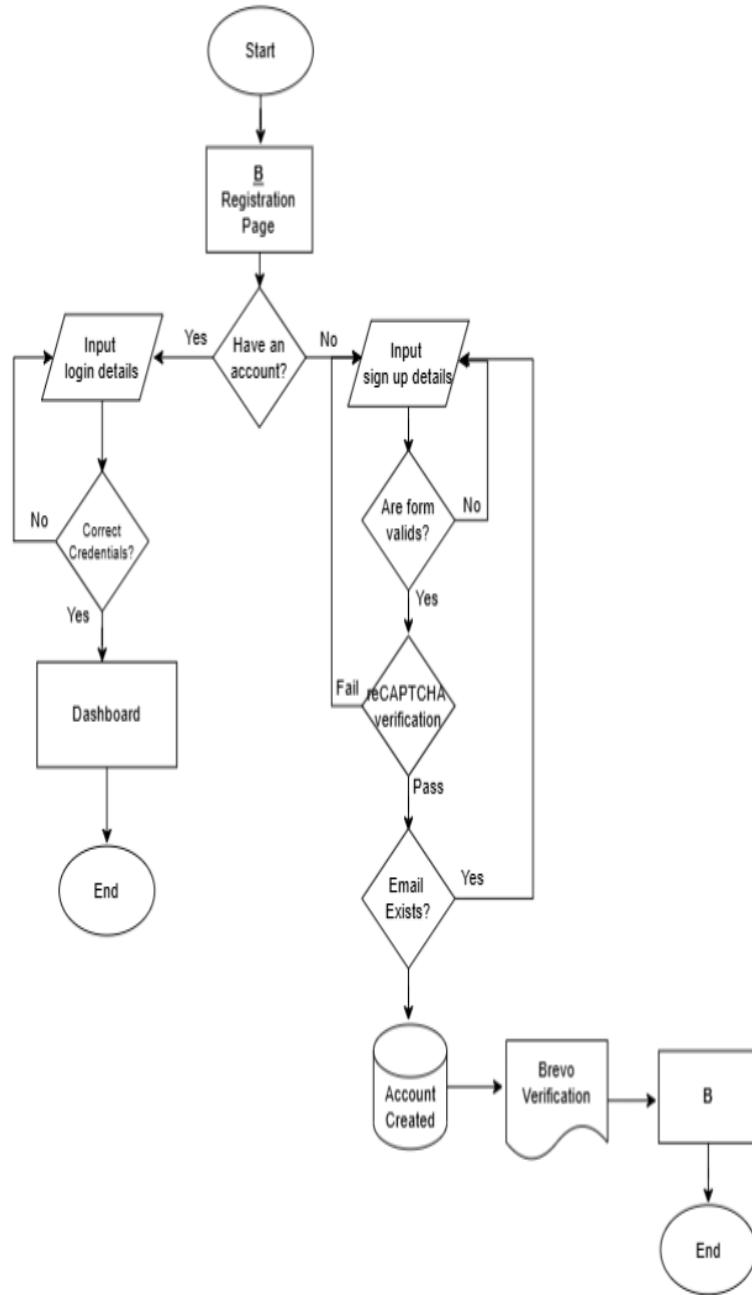


Figure 27. Client Registration Page

Figure 27 shows the registration of the client. The process starts with entering their login account if the client has already an account. If the client is a

new user, they are required to sign up before being redirect to the dashboard. The system will validate the input details, performs reCAPTCHA verification, and checks the email availability. Once verified, the account is successfully created and registration is completed.

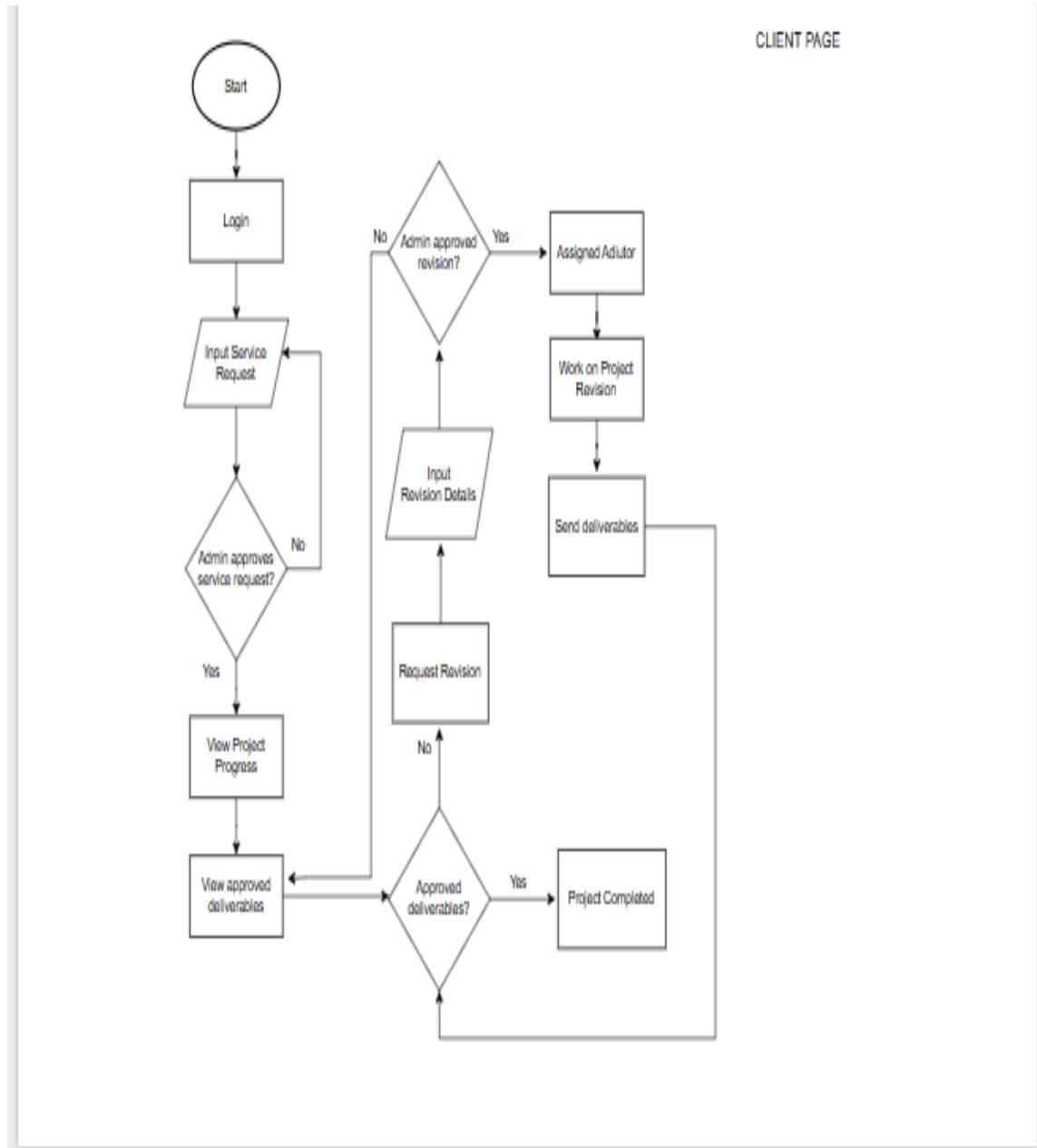


Figure 28. Client Project Monitoring

Figure 28 shows the client's interaction after logging in. The process begins when client submits a service request, and wait for the approval. Once the request was approved by the admin and done assigning an adiutor, the client will view the project progress, receive updates, and review deliverables. If the project had some errors, the revision needed and sent for another approval. In contrast, if the client approved the deliverables, the task or project will mark as completed.

Adiutor Flowchart

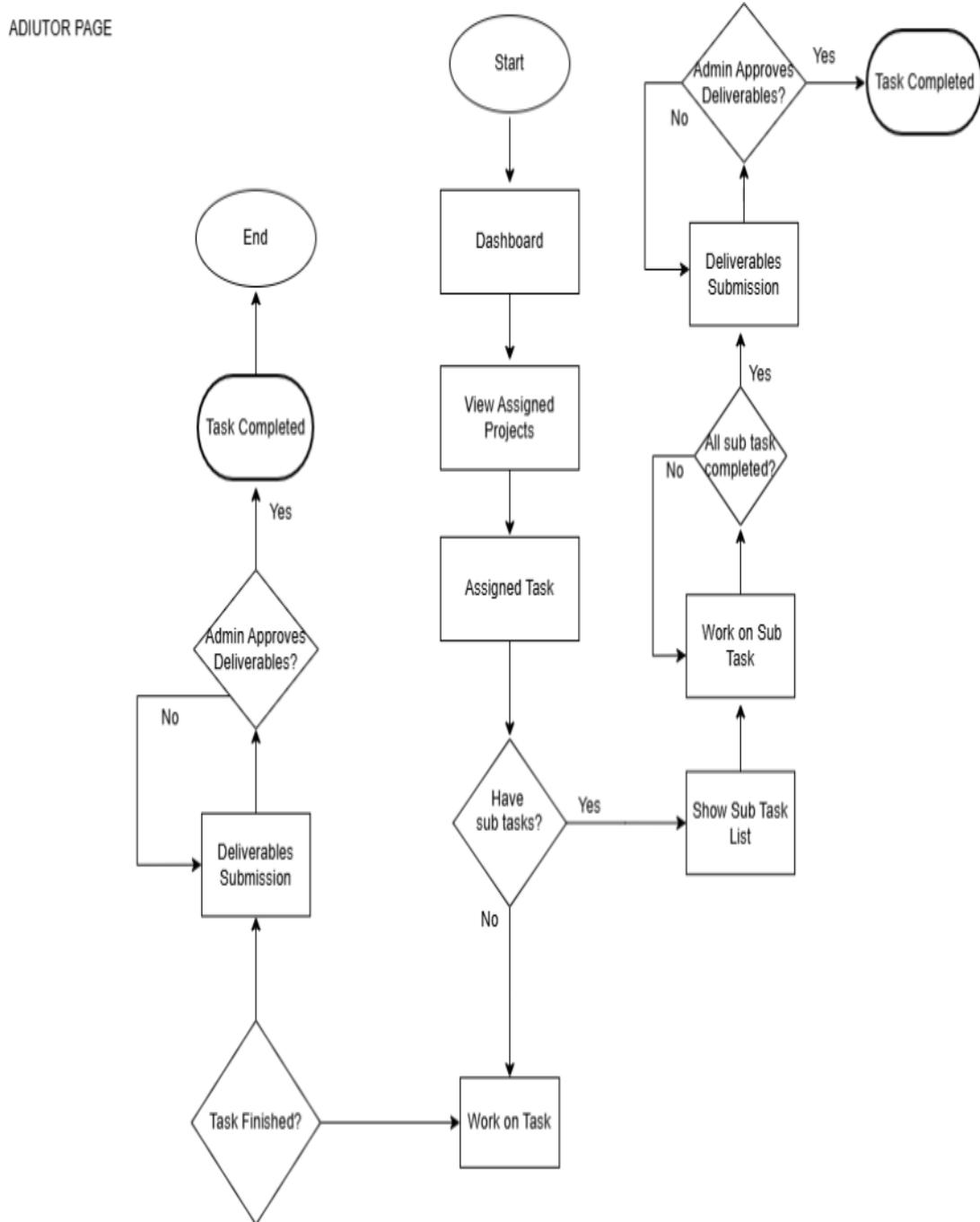


Figure 29. Adiutor Project and Task Workflow

Figure 29 shows the workflow of an adiutor, start from the task assignment to project completion. Adiutor starts from viewing assigned projects or tasks. If the

task has sub-tasks, they will work on each until completed and that will display as a progress on a task. Once the task is finished and submits deliverables for admin approval. If approved, the task will be marked as completed. Otherwise, the adiutor revises and resubmits deliverables.

Public Page

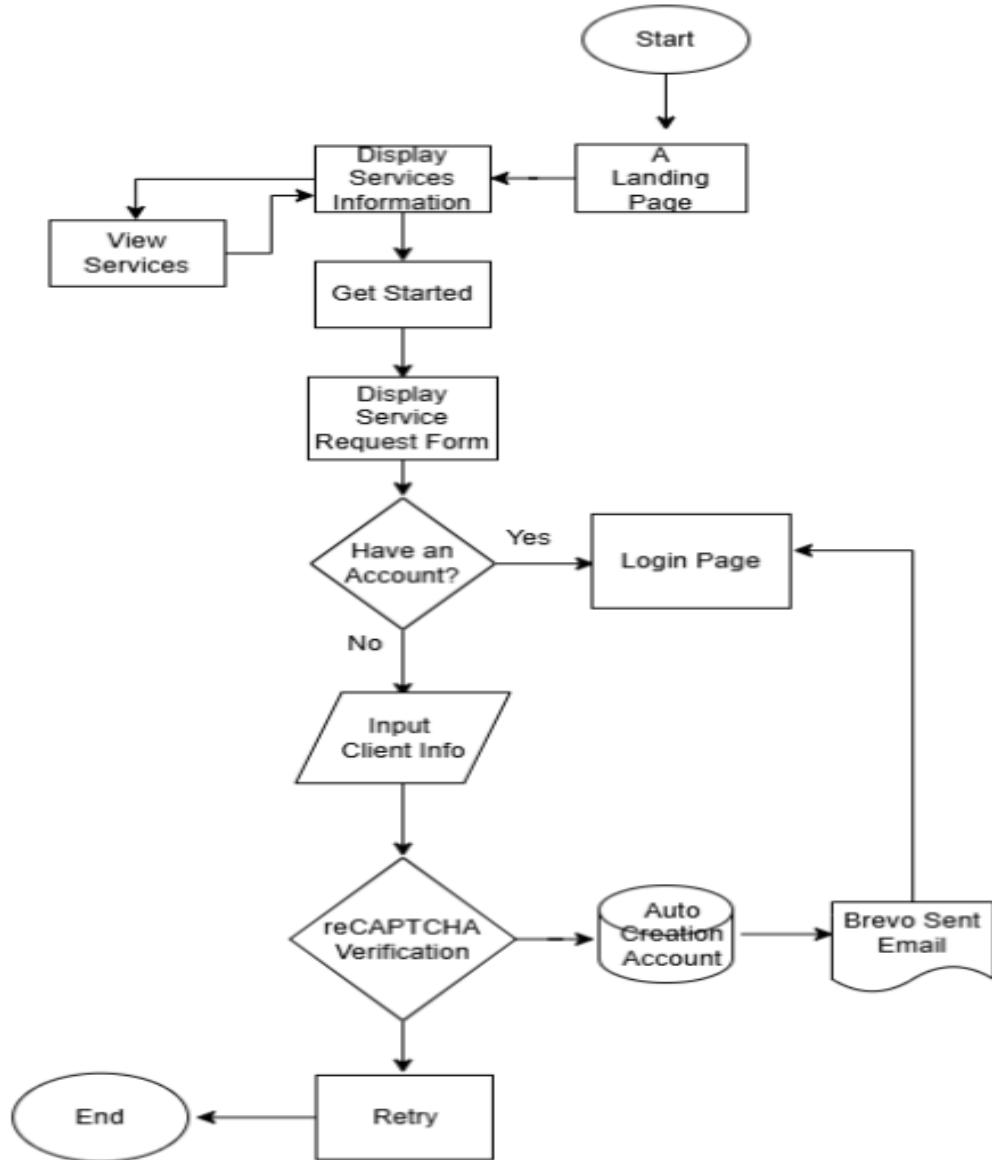


Figure 30. Public Pages Flowchart

Figure 1 shows the public page that the user encounters once they visited the website. The landing page will display including the details about the Treis Adiutor. Public users can view services, proceed to service request form, or choose to log in their account (if already have). New user can direct to the service request form, enter details, complete reCAPTCHA verification, and automatically get a created account with a confirmation email sent. This process allows them to continue with a service request process.

Entity-Relationship Diagram

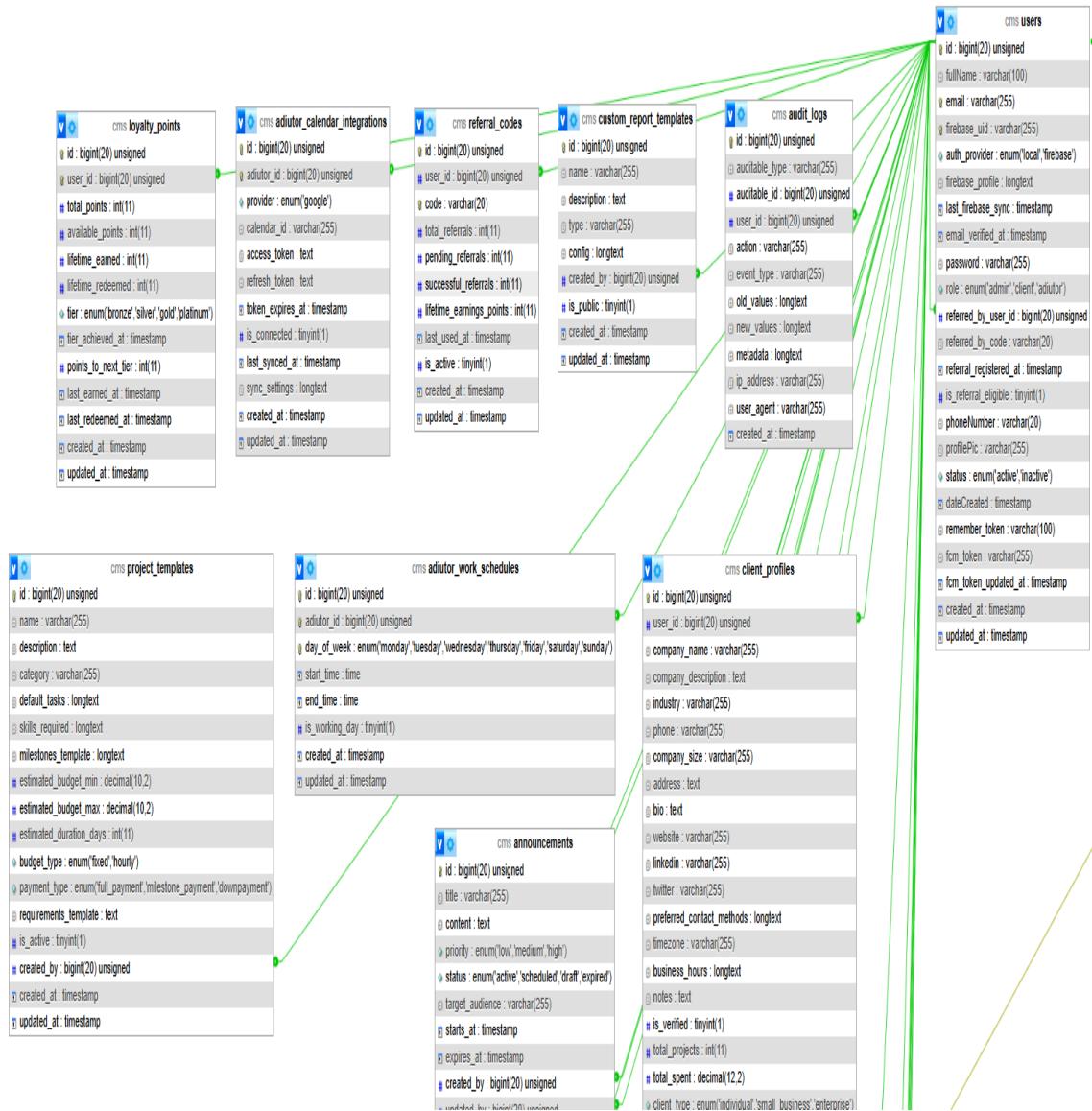


Figure 31. Close-up of the Entity-Relationship Diagram (1)

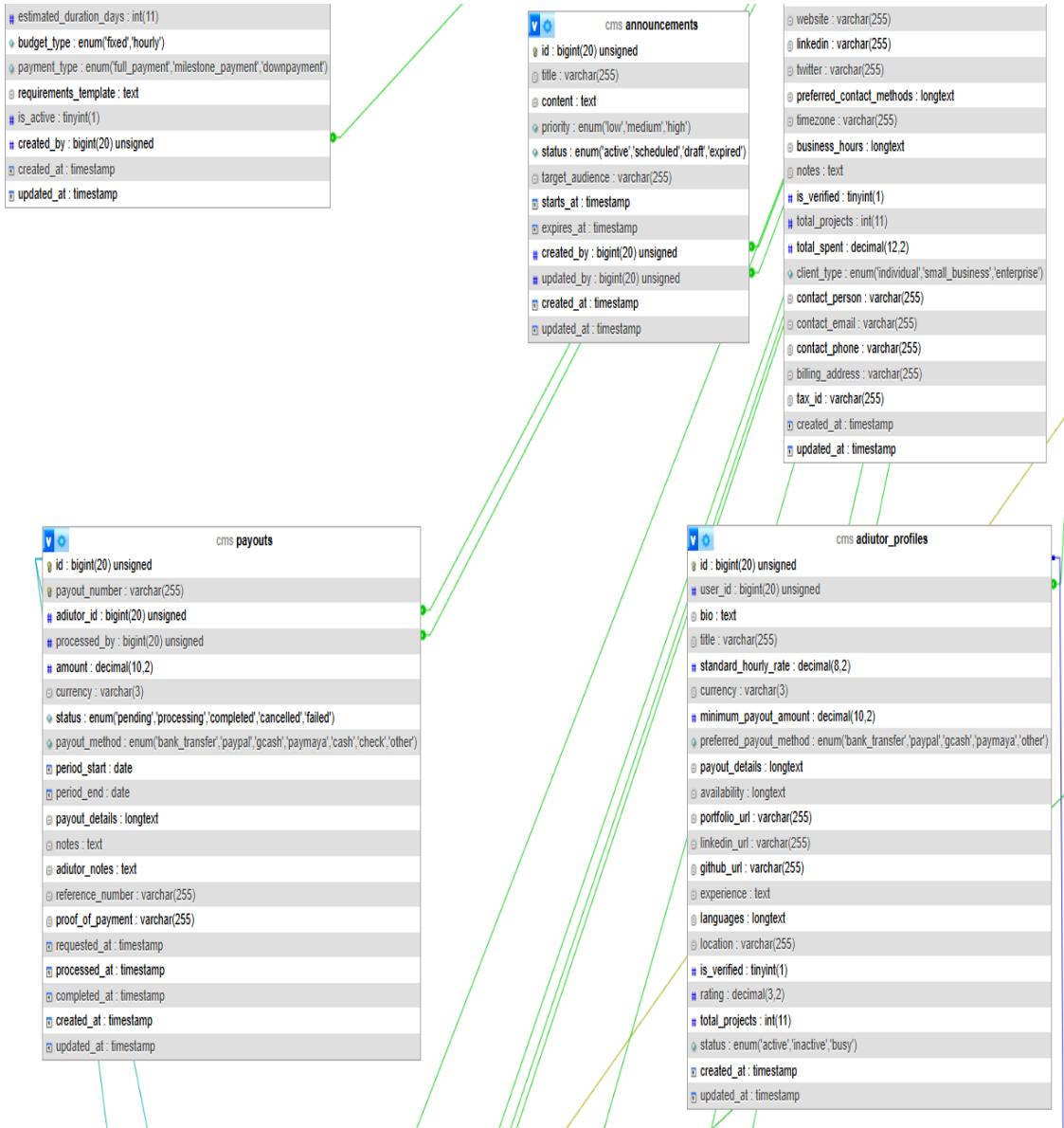


Figure 32. Close-up of the Entity-Relationship Diagram (2)

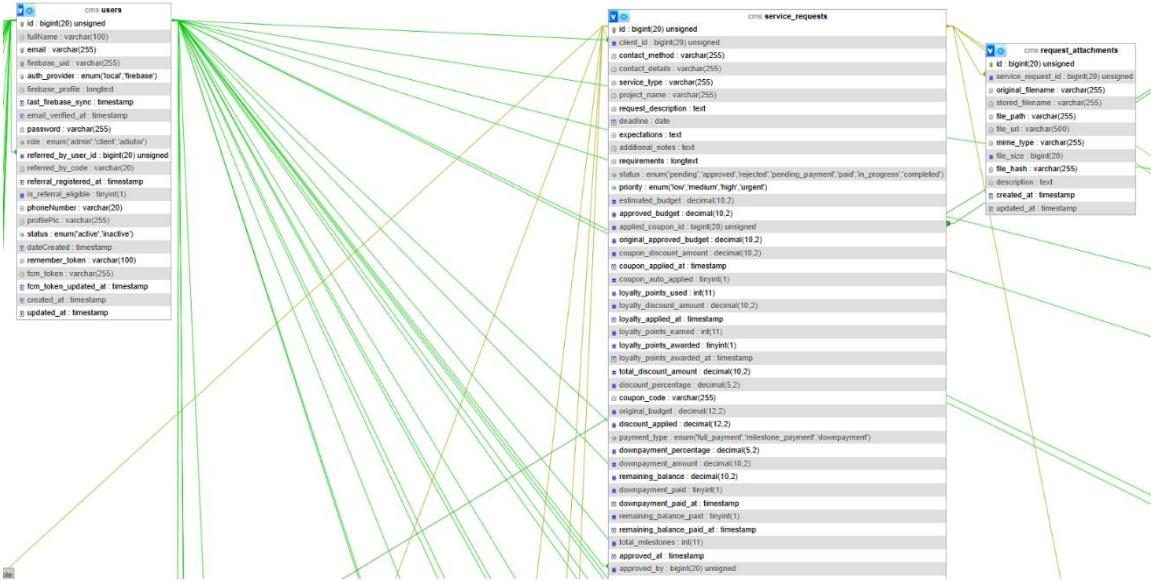


Figure 33. Close-up of the Entity-Relationship Diagram (3)

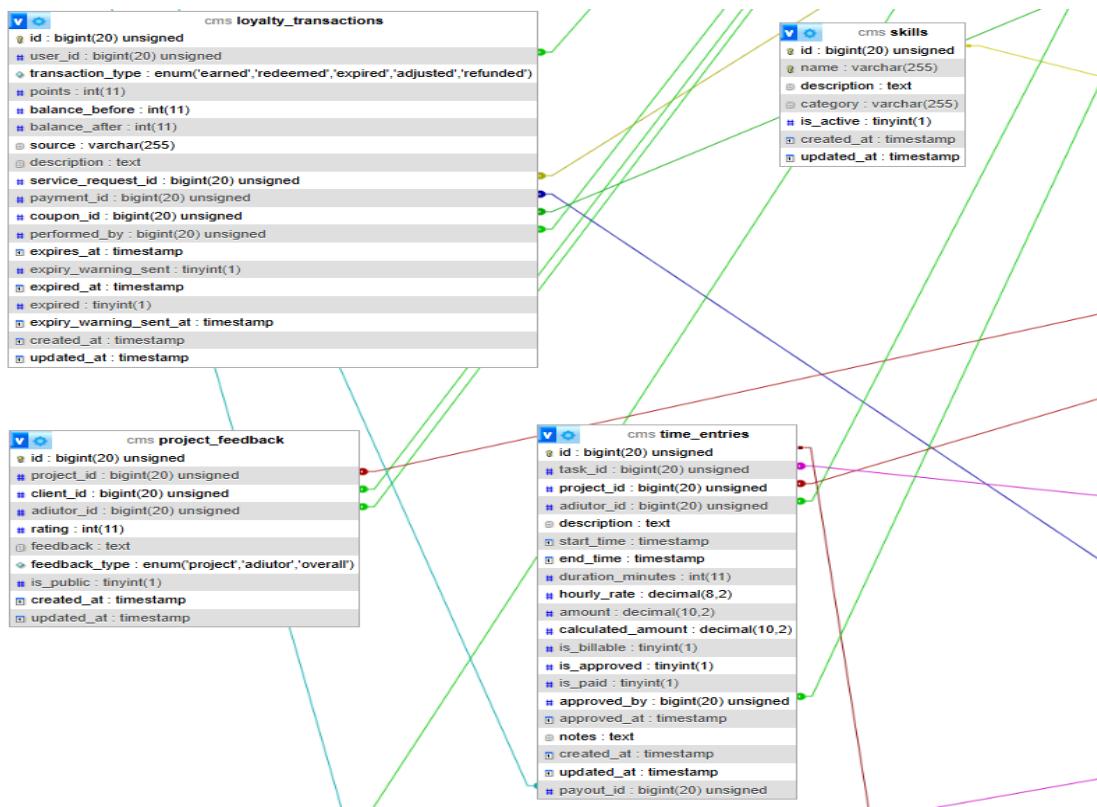


Figure 34. Close-up of the Entity-Relationship Diagram (4)

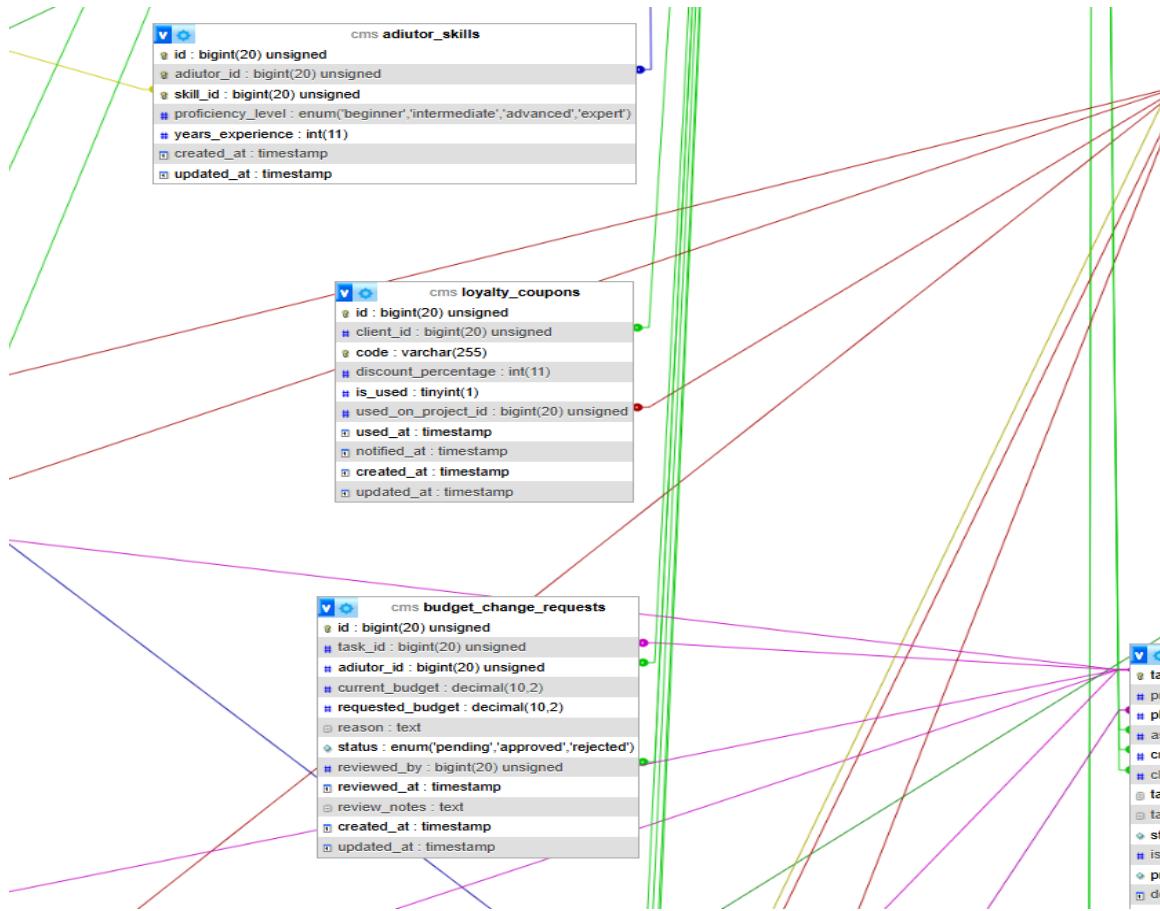


Figure 35. Close-up of the Entity-Relationship Diagram (5)

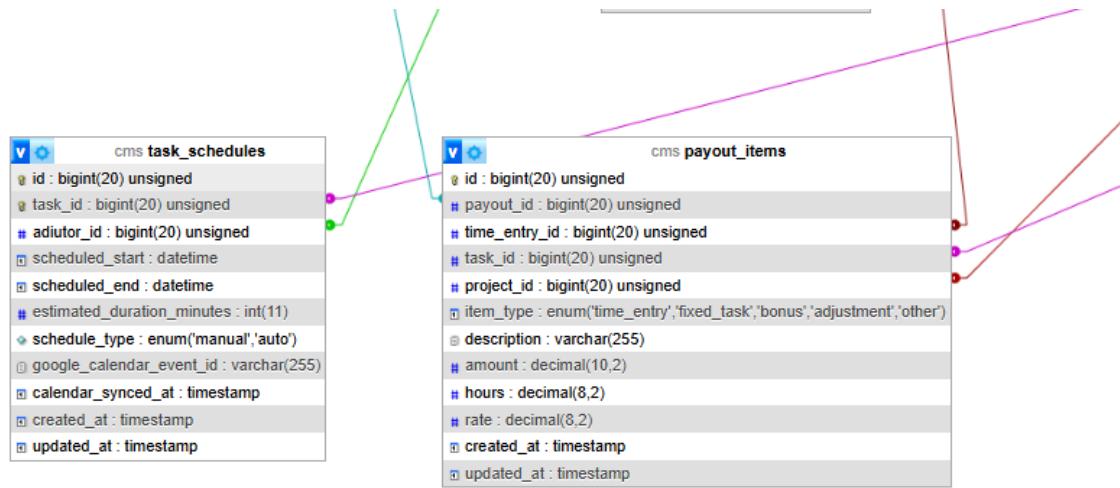


Figure 36. Close-up of the Entity-Relationship Diagram (6)

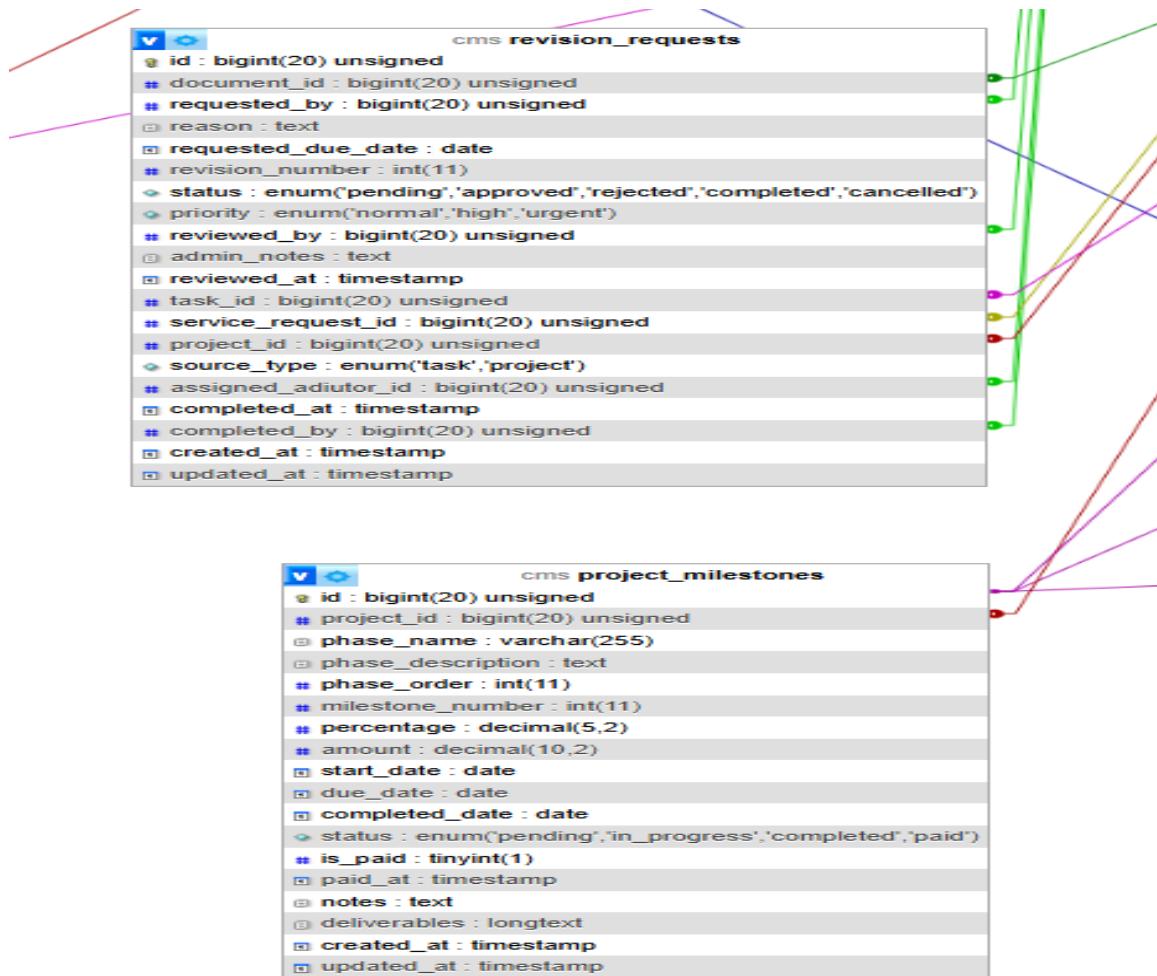


Figure 37. Close-up of the Entity-Relationship Diagram (7)

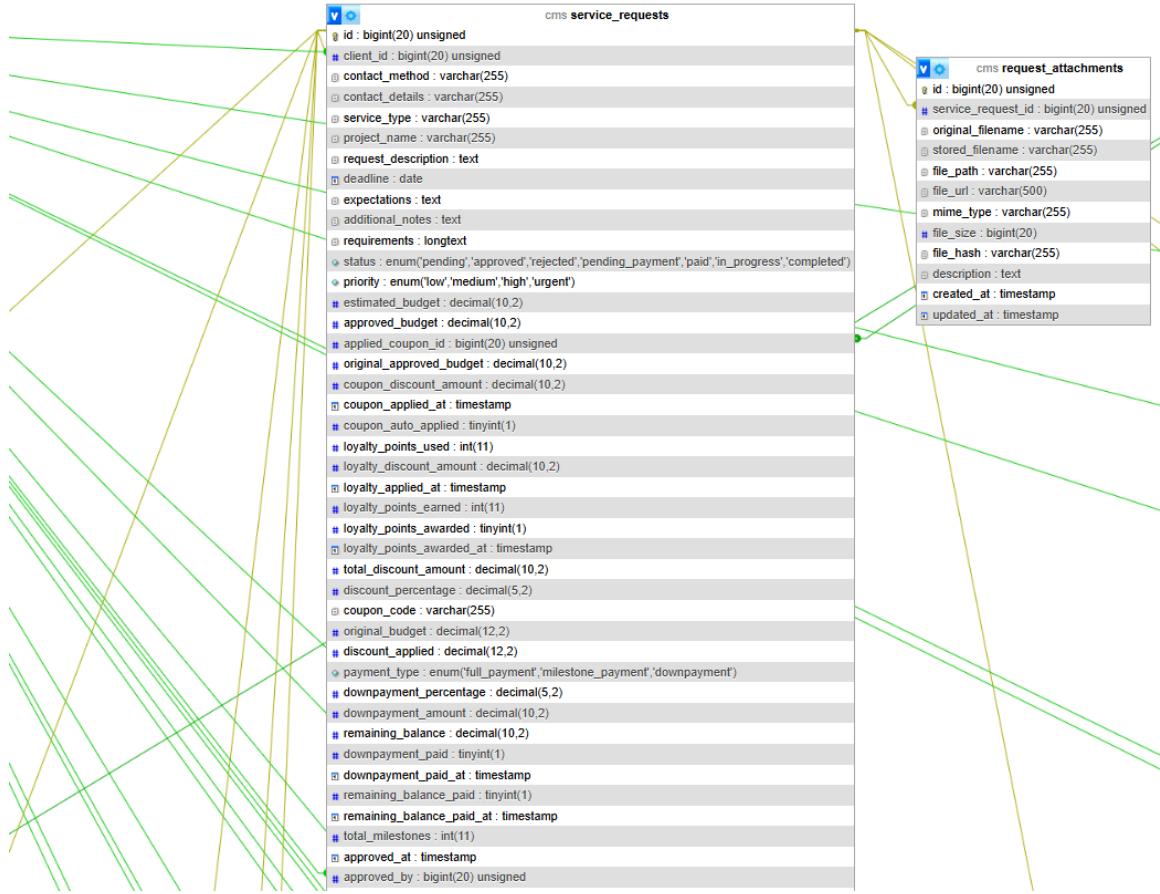


Figure 38. Close-up of the Entity-Relationship Diagram (8)

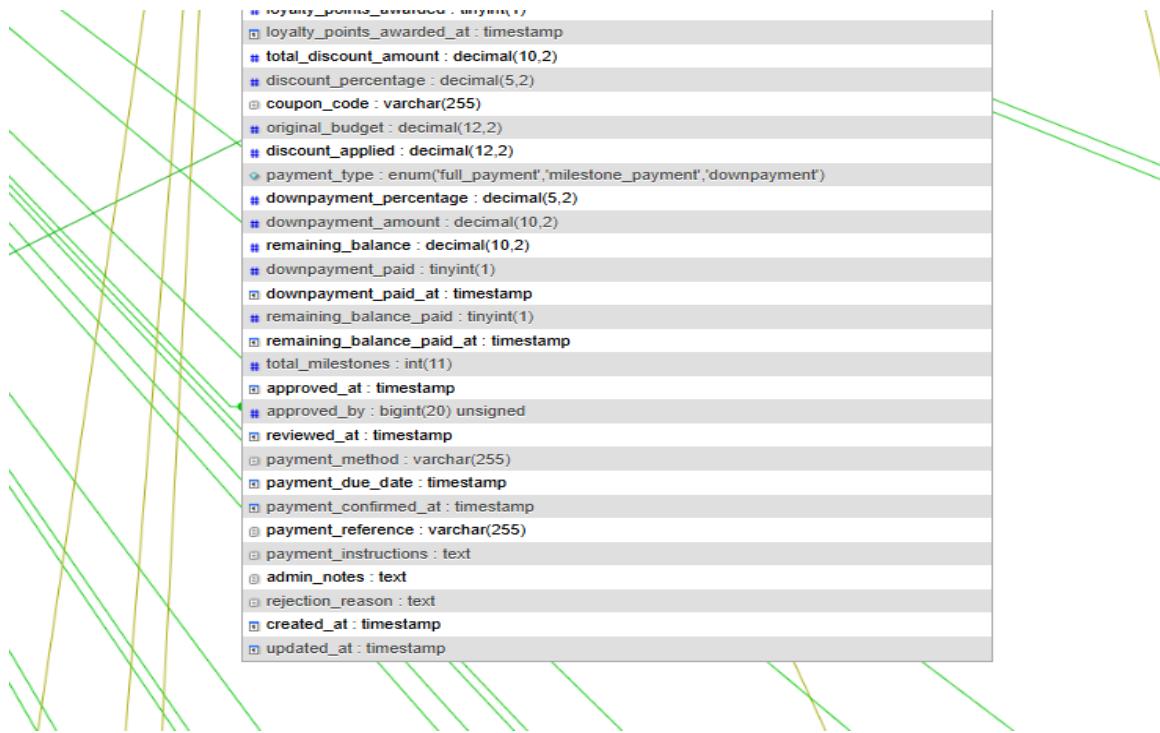


Figure 39. Close-up of the Entity-Relationship Diagram (9)



Figure 40. Close-up of the Entity-Relationship Diagram (10)

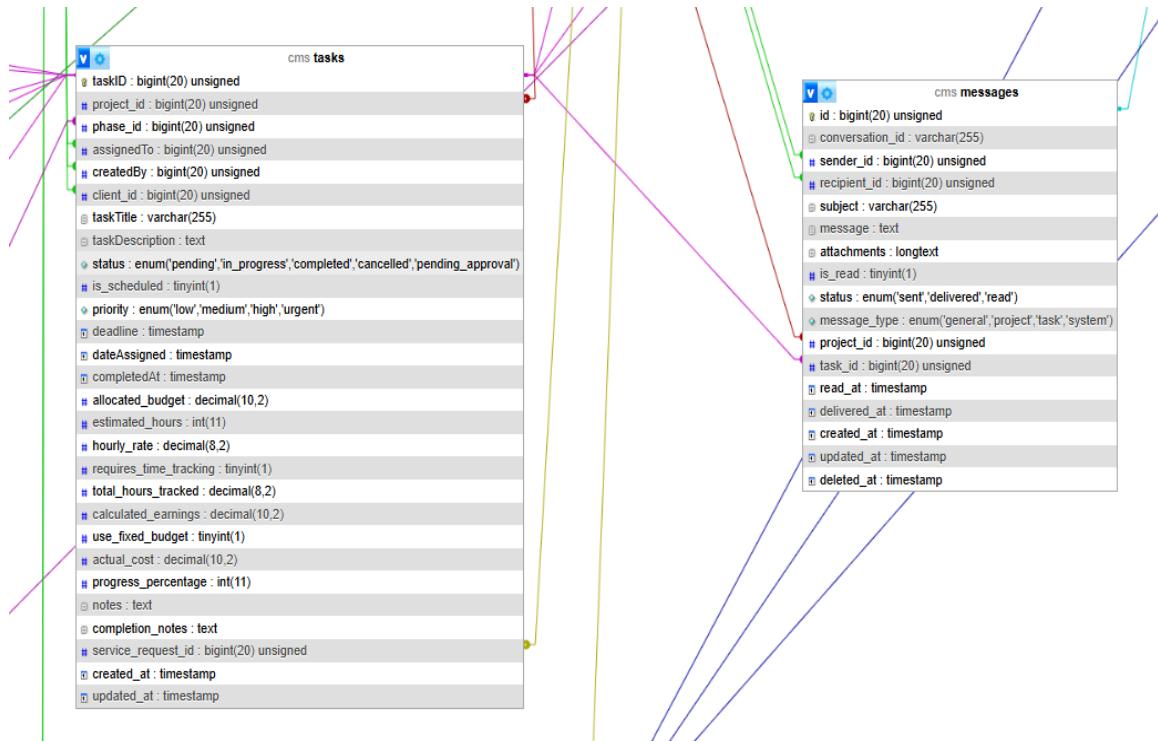


Figure 41. Close-up of the Entity-Relationship Diagram (11)

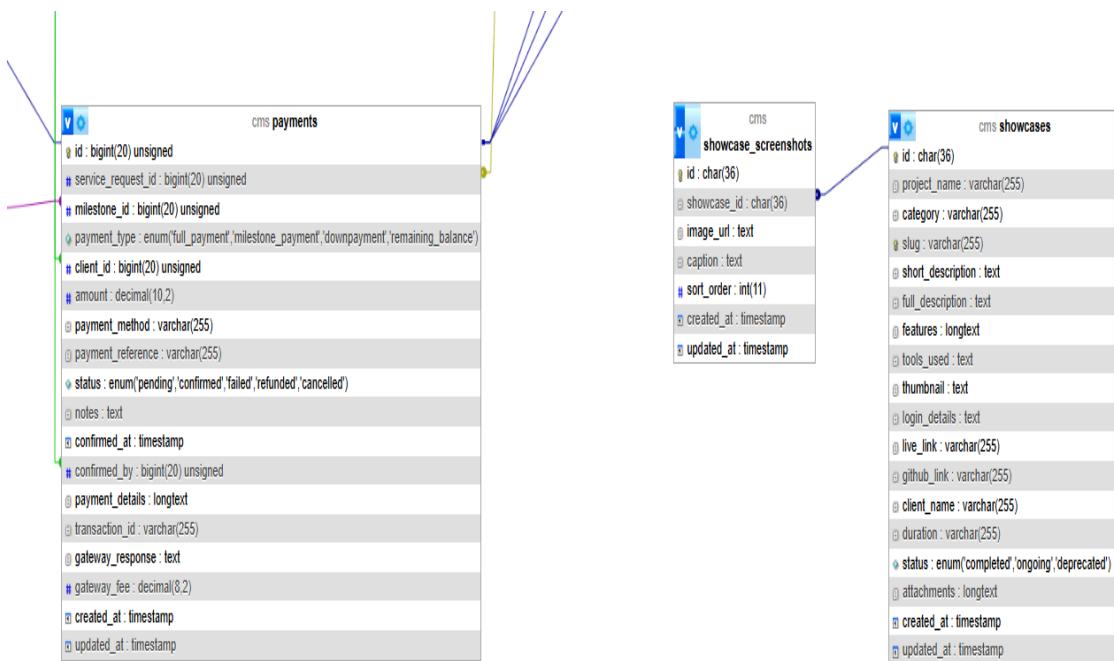


Figure 42. Close-up of the Entity-Relationship Diagram (12)

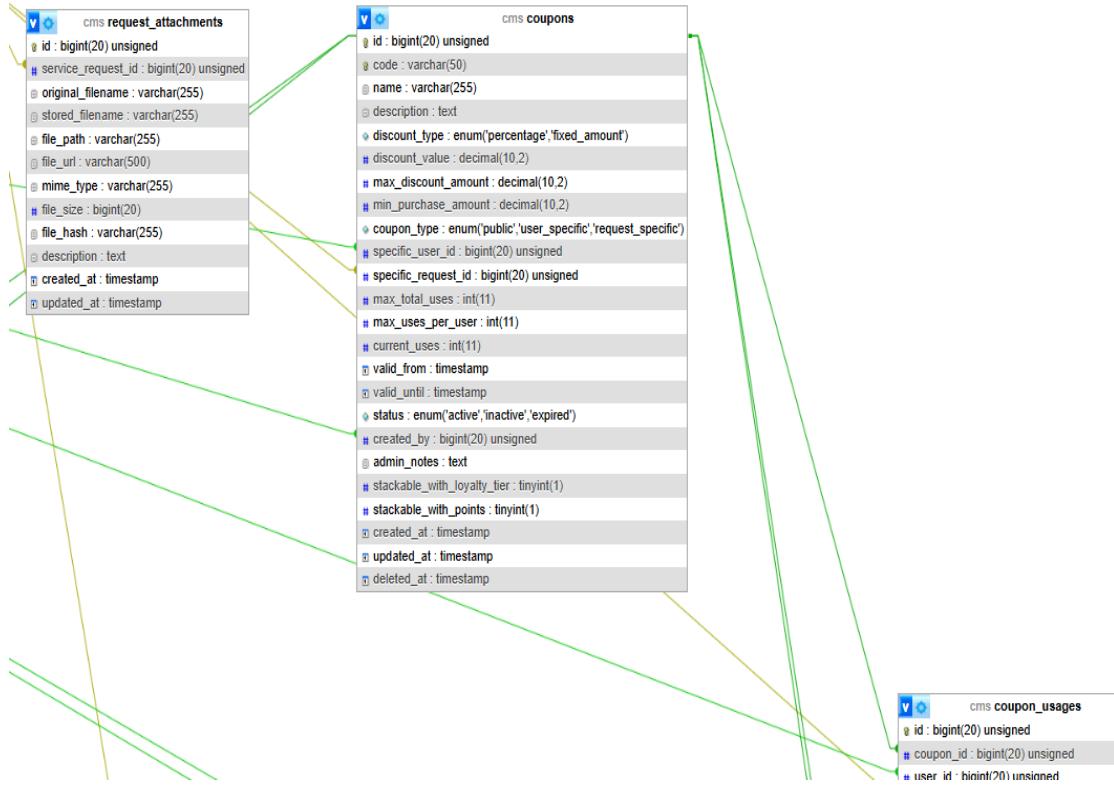


Figure 43. Close-up of the Entity-Relationship Diagram (13)

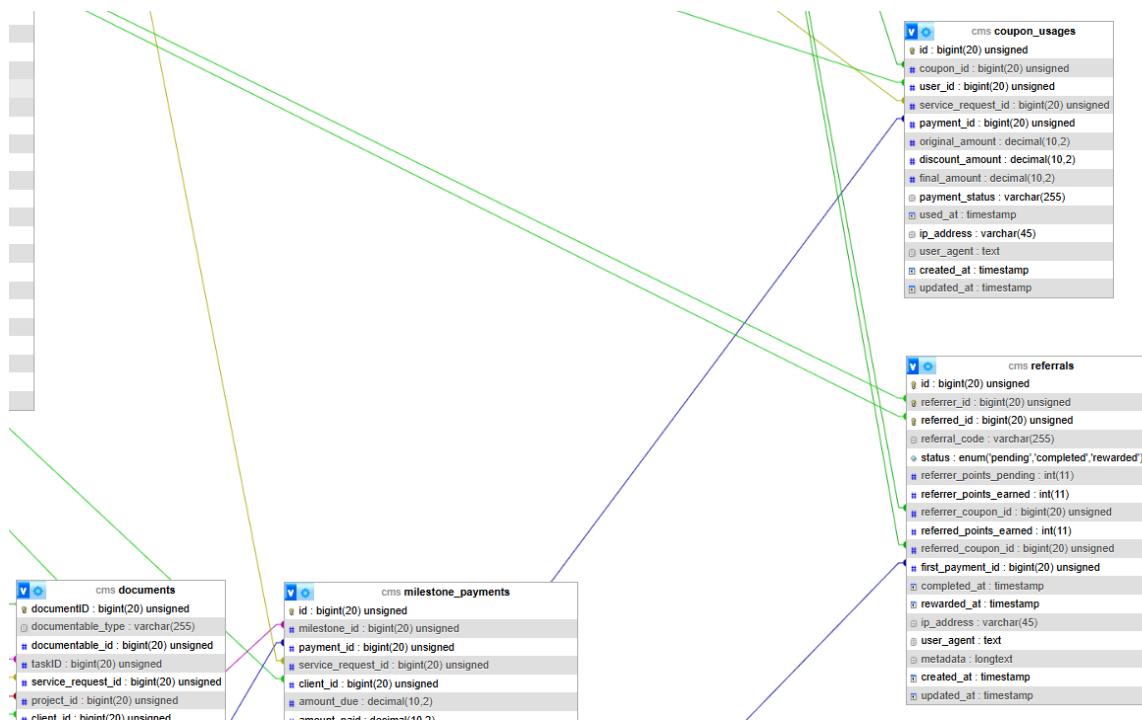


Figure 44. Close-up of the Entity-Relationship Diagram (14)

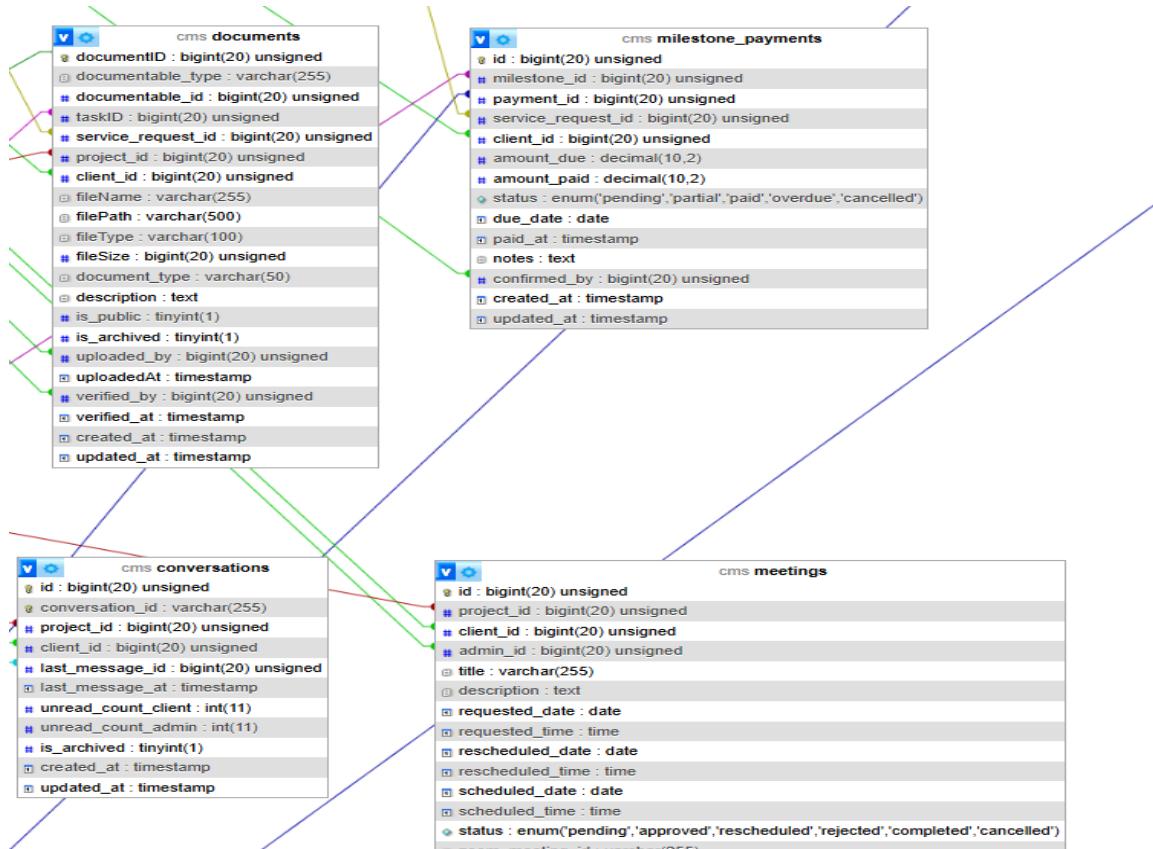


Figure 45. Close-up of the Entity-Relationship Diagram (15)

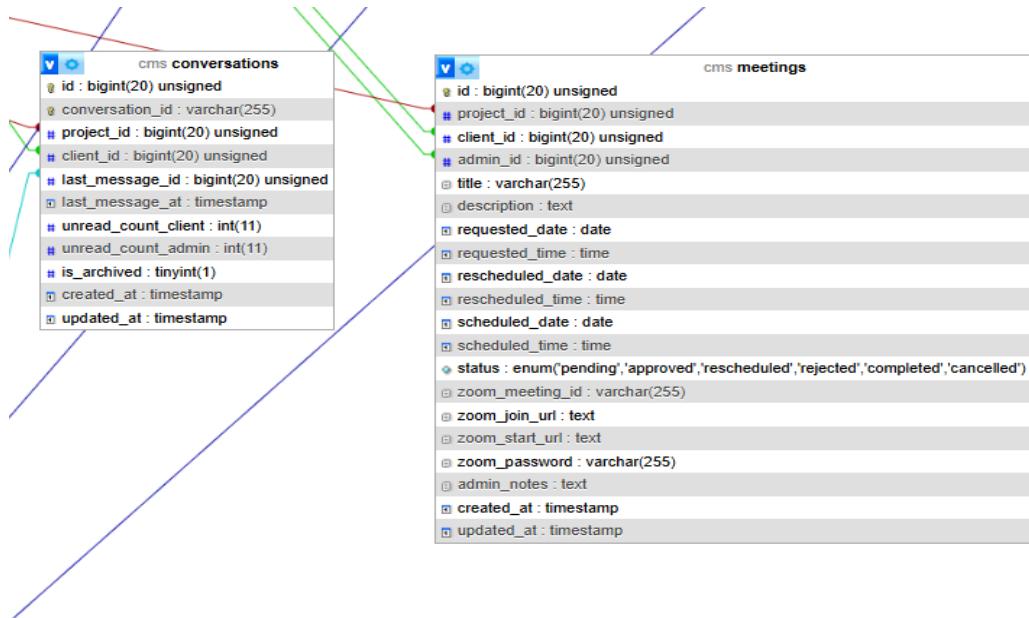


Figure 46. Close-up of the Entity-Relationship Diagram (16)

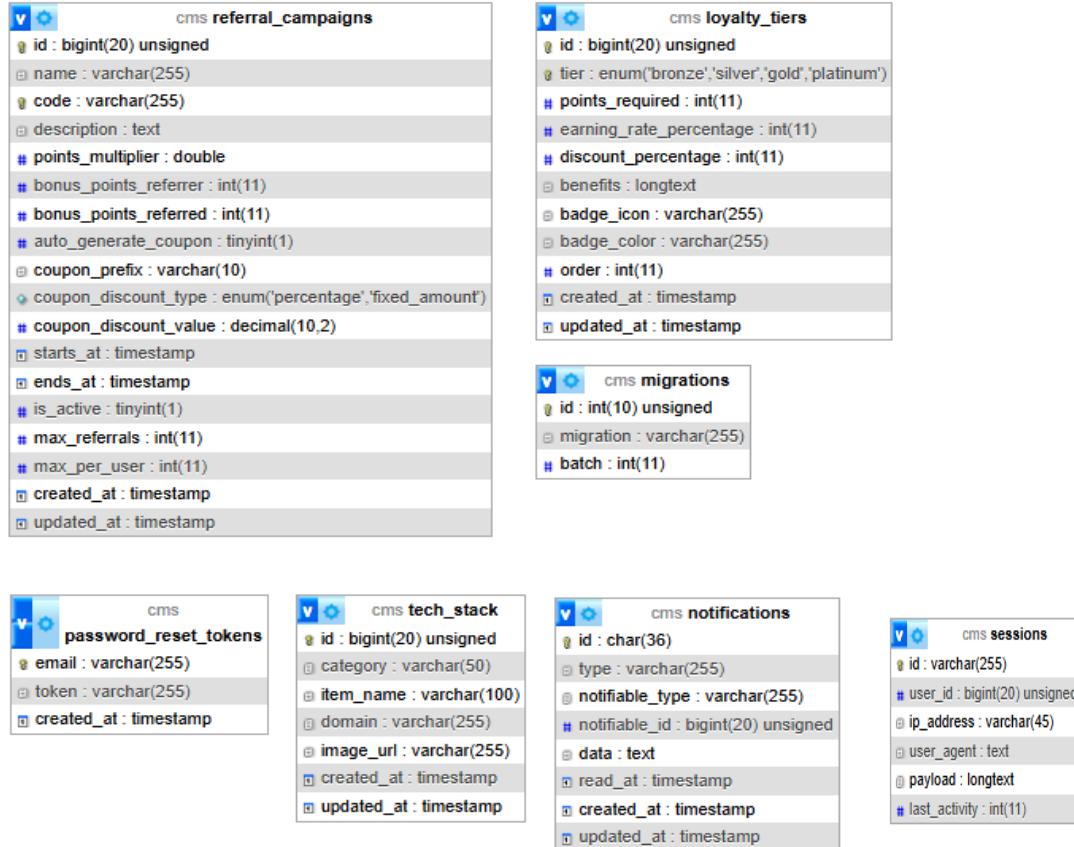


Figure 47. Close-up of the Entity-Relationship Diagram (17)

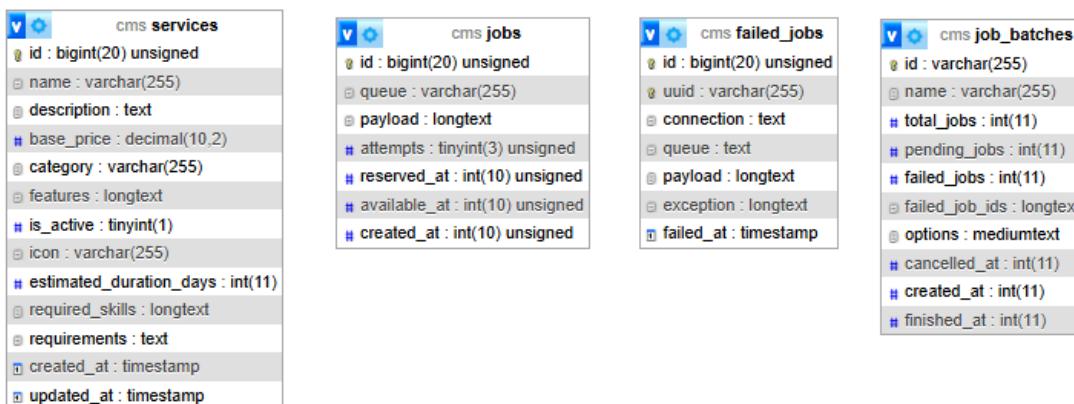


Figure 48. Close-up of the Entity-Relationship Diagram (18)

APPENDICES

**Appendix A
Approved Letter**

December 01, 2025

Hanna Faith Medil

Representative - Adiutor

Treis Adiutor

Calauan, Laguna, Philippines 4012

Subject: Request for Approval to Conduct Client–Server Technology Project

Dear Hanna Faith Medil,

Good day!

We are students from Laguna State Polytechnic University Santa Cruz Campus under the College of Computer Studies in the program Bachelor of Science in Information Technology, currently enrolled in the subject Client–Server Technology. As part of our course requirements, we are tasked to develop a functional information system or a website for a real client. This aims to help students gain hands-on experience while providing your organization with a system that may support and improve your operations.

We would like to respectfully request your approval to select **Treis Adiutor** as our official client for this project. With your permission, we will gather necessary information, conduct interviews, and study your processes to ensure that the system we develop aligns with your needs.

Below are the proposed functions and features of the Website we will create:

Proposed System/Website Name: Treis Adiutor System: A Web-Based Multi-Stakeholder Platform for Automated Service Request Processing, Project Task Management, and Integrated Financial Operations

Proposed Functions/Features:

- Feature 1 – *Authentication & Security*
- Feature 2 – *User Management*
- Feature 3 – *Service Request Management*
- Feature 4 – *Project Management*
- Feature 5 – *Task Management*
- Feature 6 – *Time Tracking & Earnings*
- Feature 7 – *Payment Management*
- Feature 8 – *Document / Deliverables Management*
- Feature 9 – *Communication Management*
- Feature 10 – *Loyalty, Coupon, & Referrals*
- Feature 11 – *Revision Management*
- Feature 12 – *Reporting & Analytics*

Should you have suggestions, revisions, or additional features you wish to include, we are very willing to incorporate them into the project.

Rest assured that all information shared with us will be treated with full confidentiality and will only be used for academic purposes.

We hope for your favorable response. Thank you very much for considering our request. We look forward to the opportunity to collaborate with your organization.

Respectfully yours,

PRINCESS ANNE AZUCENA
09750193787

SOFIA LORRAINE GONZAGA
09624626737

MA. LENA THERESE QUIZON
09491046704

MARK ANDREW SOLIMAN
09614736286

Noted by:

HARLENE GABRIELLE E. ORGINES
Subject Instructor – Client–Server Technology

Approved by:


HANNA FAITH MEDIL
Representative - Adiutor

References:

- [2] Reznikov, R. (2025). Enhancing Project Management Success through Artificial Intelligence.