**Project 2510: Technical Design Document**

**Introduction**

This document provides a comprehensive technical overview of the Flask-based web application designed for managing clients, equipment, and contracts. It outlines the system architecture, data flow, API specifications, database schema, and the structure of each application component.

**System Architecture**

**Overview**

Describe the high-level architecture of the system, including:

* Frontend components (HTML, CSS, JavaScript).
* Backend components (Flask, Python).
* Database (SQLite).

**Diagram**

Include a system architecture diagram illustrating:

* Client-server interaction.
* Frontend-backend communication.
* Database integration.

**Data Flow Diagram**

Provide a data flow diagram that shows:

* How data moves through the system.
* Interactions between different components for key operations (e.g., creating contracts, managing equipment).

**API Specifications**

**Endpoints**

List and describe all API endpoints, including:

* Endpoint URLs.
* Request methods (GET, POST, PUT, DELETE).
* Request parameters and body structure.
* Response structure and status codes.

**Authentication**

Detail any authentication mechanisms (if applicable).

**Database Schema**

**Tables**

Document each table in the database, including:

* clients: Structure, fields, and relationships.
* equipment: Structure, fields, and how it's linked to clients.
* contracts: Structure, fields, and relationships with clients and equipment.

**Diagram**

Include a database schema diagram showing:

* Tables.
* Fields.
* Relationships and foreign keys.

**Modules and Components**

**Overview**

Describe each major component or module of the application:

* Client Management Module.
* Equipment Management Module.
* Contract Management Module.
* Job Scheduling Logic.

**Details**

Provide detailed descriptions for each module, including:

* Purpose and functionalities.
* Major classes, functions, and their interactions.
* Integration points with other modules.

**Development Environment and Tools**

Detail the development environment setup, including:

* Required software and tools.
* Steps for setting up the development environment.
* Version control system and branch management strategies.

**Testing Strategy**

Outline the testing approach, covering:

* Unit testing.
* Integration testing.
* User acceptance testing.

**Deployment Plan**

Describe the deployment process, including:

* Steps for deploying the application to a production environment.
* Rollback procedures.

**Security Considerations**

Discuss security measures, such as:

* Data protection.
* User authentication

**Conclusion**

Summarize the key aspects of the technical design and its importance for the development and maintenance of the application. All technical documentation is subject to change as development progresses.