

Software Requirements Analysis and Design

This document outlines the requirements analysis and design for **Clean Maestro**, a solution tailored for operational efficiency in cleaning companies.

Index

Index.....	2
1.0 Introduction.....	3
1.1 Purpose.....	3
1.2 Scope.....	3
1.3. Glossary.....	3
2.0 System Overview.....	5
2.1 Project Perspective.....	5
2.2 System Context.....	5
General Constraints.....	5
2.3 Assumptions and Dependencies.....	5
3.0 Functional Requirements.....	6
Client Management.....	6
Task Management.....	6
Resource Management.....	6
Reporting (Auto Generated Reports).....	6
3.2 Use Cases.....	6
3.2.1 Use Case 1: Client Management (Manager).....	6
3.2.2. Use Case 2: Task Assignment (Supervisor).....	7
3.2.3. Use Case: Report Generation (System Autogeneration).....	7
3.3 Data Modelling and Analysis.....	8
3.3.1. UML Class Diagram.....	8
3.3.2. Normalized Data Model Diagram.....	9
3.3.3. Activity Diagrams (Danny).....	10
3.3.4. Sequence Diagrams.....	14
3.4 Process Modelling.....	16
3.4.1. Data Flow Diagram.....	16
4.0 Non-Functional Requirements.....	16
Performance.....	16
Reliability.....	16
Security.....	16
Maintainability.....	16
Portability.....	17
5.0 Logical Database Requirements.....	17
6.0 Other Requirements.....	17
7.0 Approval.....	17

1.0 Introduction

The Introduction section provides an overview of the system using software requirements analysis and design for the scope of the system.

1.1 Purpose

This Software Requirements Specification (SRS) details the high-level requirements for Clean Maestro, which addresses management challenges for cleaning companies. This document is intended for both stakeholders and developers, defining what the system will achieve and the constraints under which it will operate. This serves as a foundation for developers and aligns stakeholder expectations.

1.2 Scope

Clean Maestro is a software solution that will enable cleaning companies to manage their staff, clients, resources, and tasks more effectively. The application provides a centralized platform to handle scheduling, real-time task tracking, inventory management, and reporting.

- **In Scope:**
 - **Client management:** Add, view, and remove clients, as well as set client-specific cleaning requirements.
 - **Task management:** Assign tasks to staff members, monitor task completion, and track task status.
 - **Resource management:** Track and manage cleaning supplies and tools.
 - **Reporting:** Generate client-specific reports and performance metrics for internal review.
 - **Role-based access:** Ensure secure access based on user roles (e.g., managers, staff, clients).
 - **Real-time notifications:** Provide alerts for overdue tasks, upcoming shifts, and incomplete work.
- **Out of Scope:**
 - **Financial tracking and payroll systems:** Not within the scope of this system.
 - **Integration with external non-cleaning services:** Limited to cleaning-specific operations and requirements.

1.3. Glossary

Term	Definition
Client	An institution that outsources cleaning services to a cleaning company. Examples include hotels, casinos, and theater venues.
Cleaning company	An organization responsible for performing cleaning services. In some use cases, a Client may also act as its own Cleaning Company, depending on the business logic within Clean Maestro.
Owner	The proprietor of the Cleaning Company. In some scenarios, the Owner may also serve as part of the staff, such as a <i>Manager</i> or <i>Supervisor</i> .
Manager	A member of the managerial staff responsible for overseeing operations within the <i>Cleaning Company</i> . Duties include assigning <i>Supervisors</i> and <i>Cleaning Staff</i> and setting up specific cleaning <i>requirements</i> .
Supervisor	A member of the managerial staff responsible for managing logistics, coordinating on-site cleaning operations, and supervising <i>Cleaning Staff</i> .
Cleaning staff	Team members tasked with performing cleaning <i>Tasks</i> as assigned by the <i>Supervisor</i> or <i>Manager</i> .
Task	A cleaning activity that includes specific steps and instructions. <i>Example: "Clean room 1017."</i>
Job	A collection of <i>Tasks</i> assigned to be

	completed within a designated time frame, often following a specific sequence. <i>Example: "Clean floor 10."</i>
Requirements	A collection of Jobs requested by a specific Client, detailing the scope and specifications of the cleaning services needed.

2.0 System Overview

The System Overview section introduces the system context and design.

2.1 Project Perspective

Clean Maestro is a **new, self-contained system** developed to fill an operational gap in the cleaning services sector. Unlike generic management solutions, Clean Maestro focuses on providing tools tailored to cleaning operations, such as task assignment, resource tracking, and real-time updates.

•

2.2 System Context

The system addresses strategic needs in the cleaning industry by streamlining processes for cleaning service providers and enhancing client satisfaction through better transparency and service customization. Clean Maestro's primary target users include cleaning companies, cleaning staff, and clients such as hotels, restaurants, and other commercial establishments.

General Constraints

- **Data Privacy and Security Compliance:** Must comply with GDPR and CCPA standards for data handling.
- **Internet Access:** Real-time updates and notifications require a stable internet connection.
- **Project Budget and Timeline:** Feature additions and scope changes will be limited by the established budget and the deadline (March 28, 2025).

2.3 Assumptions and Dependencies

Assumptions:

- Users (supervisors, managers, and staff) have basic familiarity with mobile and web applications.
- All users have access to compatible devices and a stable internet connection.

Dependencies:

- Reliable internet access is required for real-time task updates.
- Any changes to requirements during development are subject to stakeholder approval.

3.0 Functional Requirements

Client Management

- **Inputs:** Client details (e.g., name, contact info, specific cleaning needs).
- **Processing:** Storing, updating, and displaying client data.
- **Outputs:** Client lists, cleaning requirements, and client-specific reports.

Task Management

- **Inputs:** Task descriptions, assigned staff, deadlines.
- **Processing:** Assigning, updating, and tracking tasks.
- **Outputs:** Real-time notifications on task status, completion updates.

Resource Management

- **Inputs:** Resource type (e.g., cleaning supplies, tools), quantity, status.
- **Processing:** Managing inventory levels, updating resource availability.
- **Outputs:** Resource status, alerts for low-stock items.

Reporting (Auto Generated Reports)

- **Inputs:** Task status, media (e.g., images), annotations, and timestamps.
- **Processing:** At the end of each Job, the system collates all relevant progress data and formats it into a structured report.
- **Outputs:** Autogenerated Job report for the client, manager, and owner (if different), accessible through their dashboards.

3.2 Use Cases

Some use cases may include:

3.2.1 Use Case 1: Client Management (Manager)

- **Actor:** Manager
- **Preconditions:** The Manager is authenticated and has access to the client management module.
- **Basic Flow:**
 1. The Manager selects the option to add a new client.

2. The Manager inputs client details (e.g., name, address, contact details, cleaning requirements).
 3. The system saves the new client's information in the database.
 4. The system confirms that the client has been successfully added.
 5. The system autogenerates credentials for client access to the web dashboard.
- **Postconditions:** The new client's data is stored and accessible in the system.
 - **Exceptions:** If required fields are missing, the system will prompt the Manager to complete all fields.

3.2.2. Use Case 2: Task Assignment (Supervisor)

- **Actor:** Supervisor
- **Preconditions:** The Supervisor is logged in and has access to the task assignment interface.
- **Basic Flow:**
 1. The Supervisor selects a task and assigns it to a staff member.
 2. The system updates the task's status to "Assigned" and logs the assignment details.
 3. The system sends a notification to the assigned staff member with task details.
- **Postconditions:** The task is updated with assigned staff, and the status is visible to both the Supervisor and staff member.
- **Exceptions:** If the staff member is unavailable, the system will notify the Supervisor and suggest other available staff.

3.2.3. Use Case: Report Generation (System Autogeneration)

- **Actor:** System (Autogeneration Process)
- **Preconditions:** A Job is marked as completed, and all associated tasks have been updated with final statuses, photos, and annotations.
- **Basic Flow:**
 1. The system detects the completion of a Job.
 2. The system compiles data related to the Job (e.g., task statuses, images, annotations).
 3. The system formats the data into a comprehensive report.
 4. The system saves the report and grants access to the relevant users (client, manager, and owner).
- **Postconditions:** The completed report is accessible to the client, manager, and owner via their dashboards.
- **Exceptions:** If any task lacks final updates, the system prompts the responsible staff to complete reporting before generating the final report.

3.3 Data Modelling and Analysis

3.3.1. UML Class Diagram

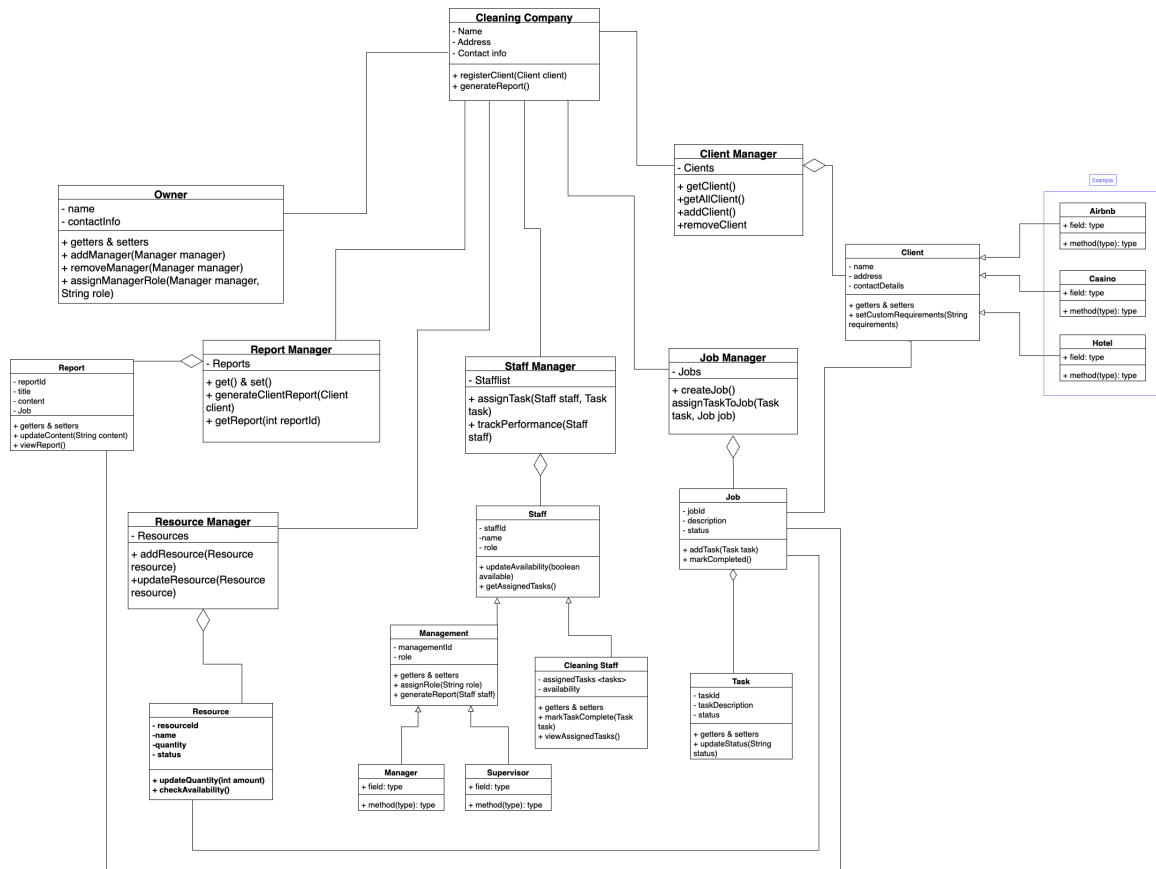


Figure 1. T50_UML_Class_Diagram.png

Note: Classes belonging to 'Client' e.g. *Airbnb*, *Casino*, *Hotel*; as well as classes belonging to 'Management' e.g. *Manager* and *Supervisor* serve as examples of the type of classes that can be registered in the App, therefore the detailed attributes and methods were not necessary to declare here.

Also, it is assumed that every class has their own `getter()` and `setter()` methods, therefore it is simply declared as `getters & setters` in the UML.

3.3.2. Normalized Data Model Diagram

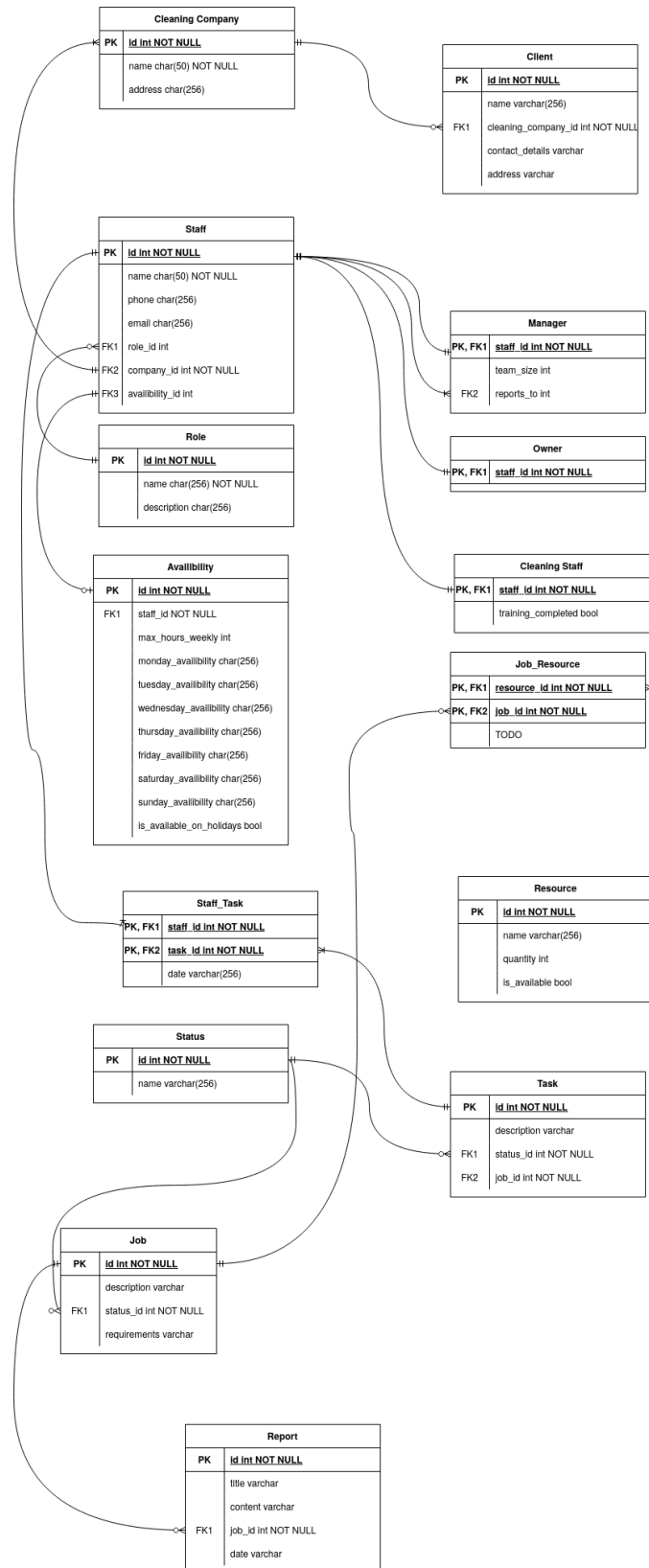


Figure 2. T50_Data_Model_Normalized.png

3.3.3. Activity Diagrams (Danny)

Log-in of registered user:

Login and Authentication

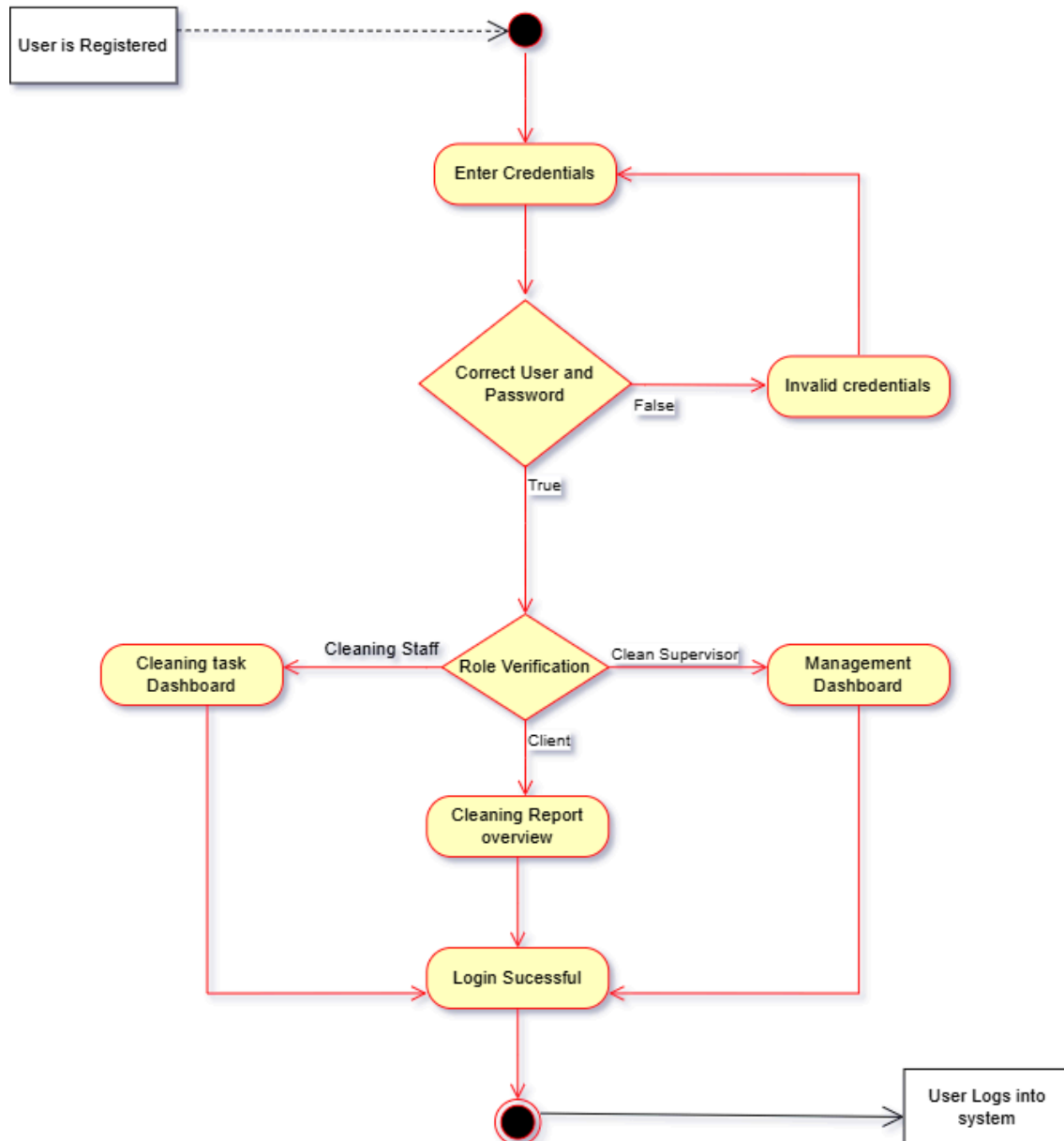
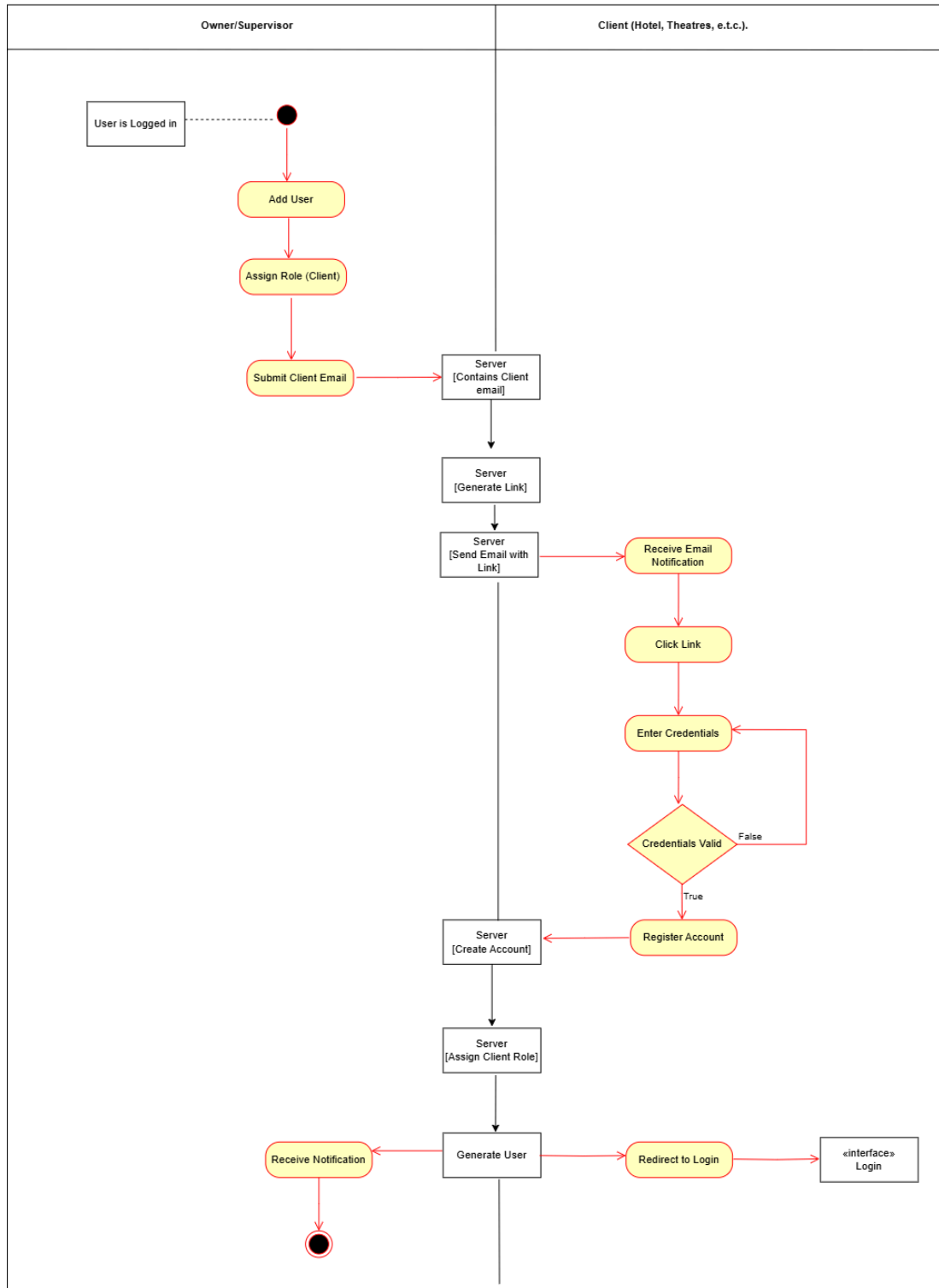


Figure 3. T50_Activity_Diagram_Login.png

Register a Client:**Client Registration and Onboarding Process****Figure 4.** T50_Activity_Diagram_Client_Registration.png

Assign a task:

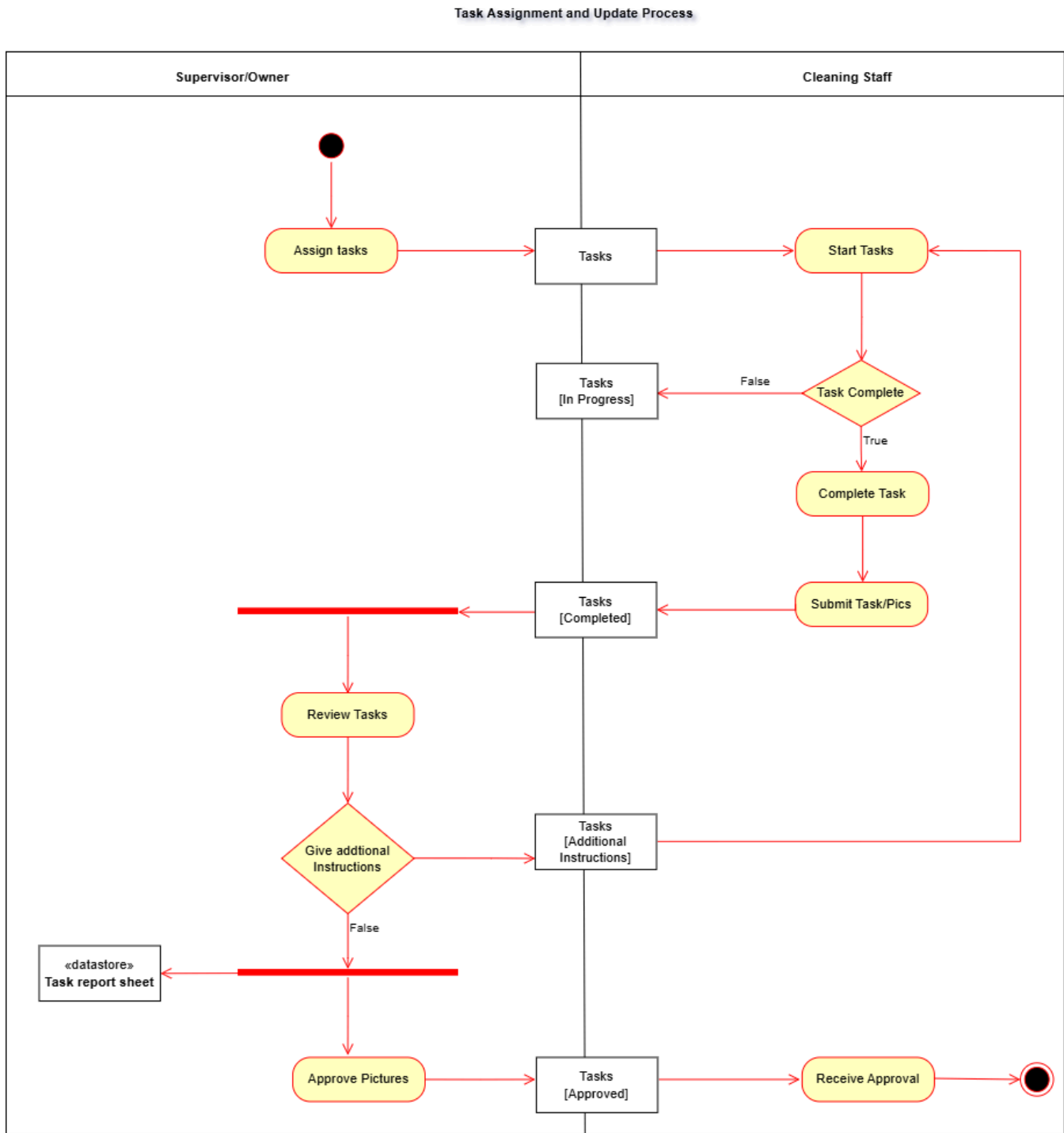
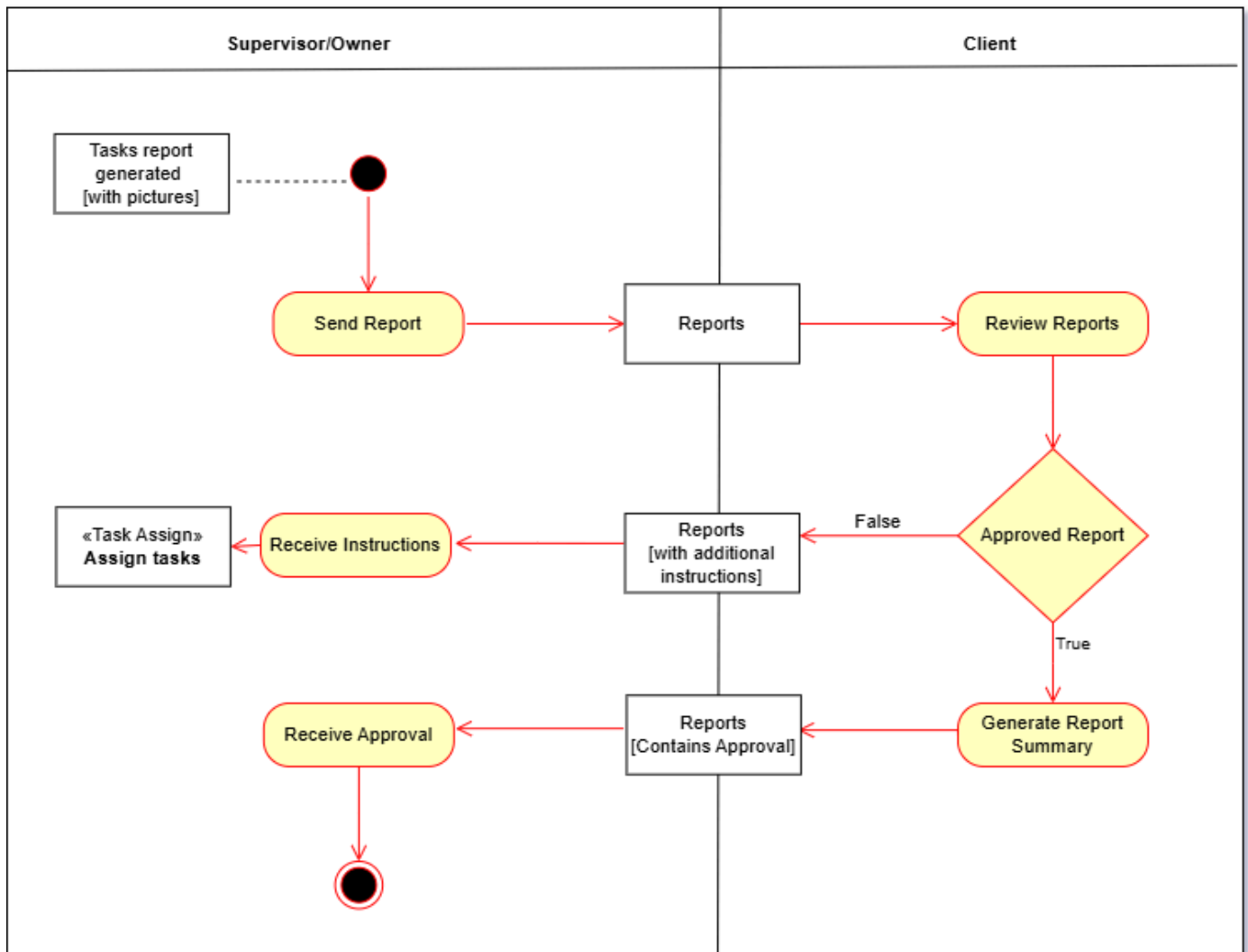


Figure 5. T50_Activity_Diagram_Assign_Task.png

Task management and report workflow:**Task Management and Reporting Workflow***Figure 6. T50_Activity_Diagram_Report_Workflow.png***Notes:**

This diagram shows the workflow after tasks are completed, this report is delivered to the client and after being accepted, the summary of the Job is autogenerated.

3.3.4. Sequence Diagrams

General Process of the Application:

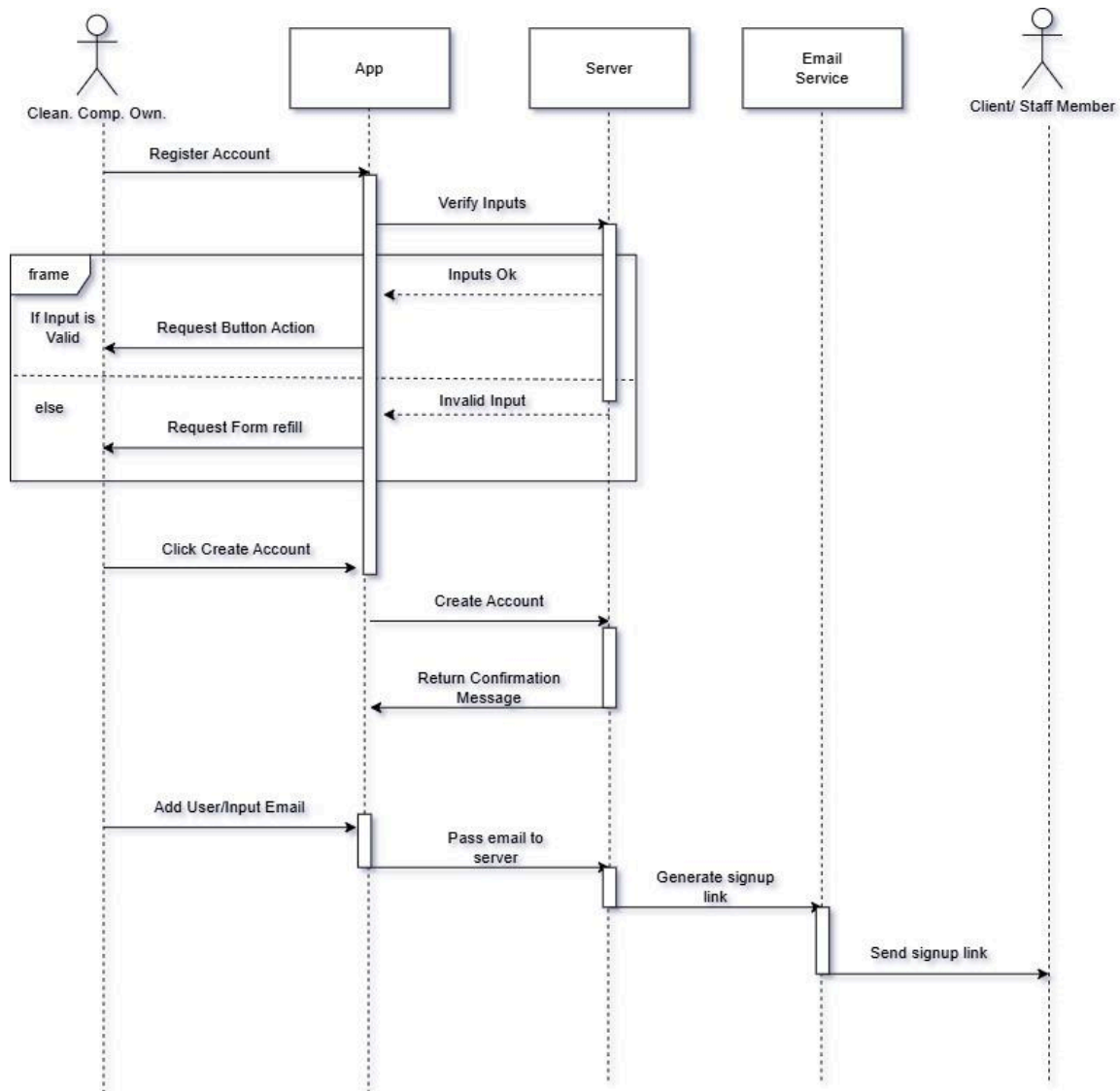


Figure 7. T50-Sequence_Diagram_General_App_Process.jpg

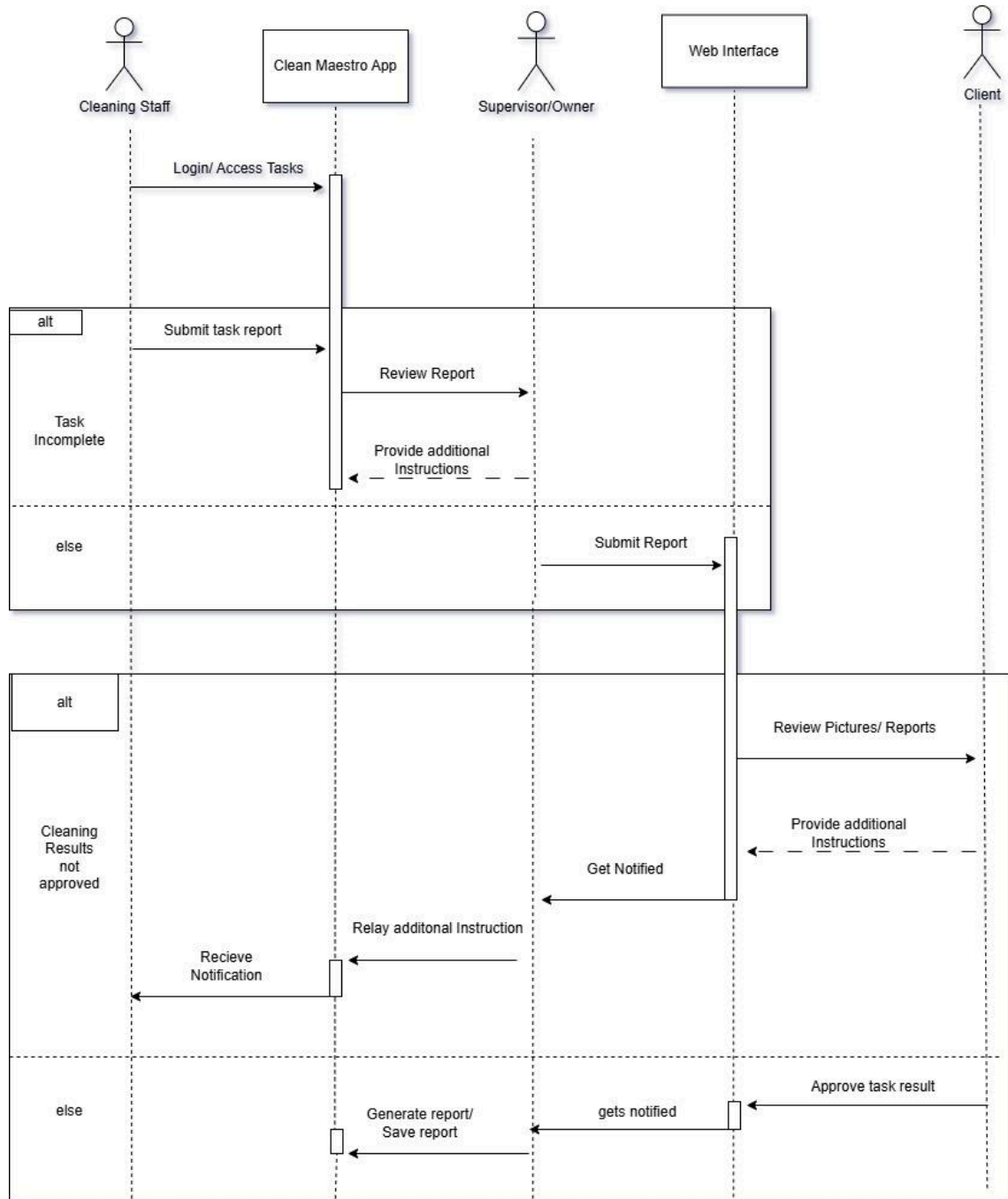
Authentication Process:

Figure 8. T50_Sequence_Diagram_Aunthentication_Process.jpg

3.4 Process Modelling

3.4.1. Data Flow Diagram

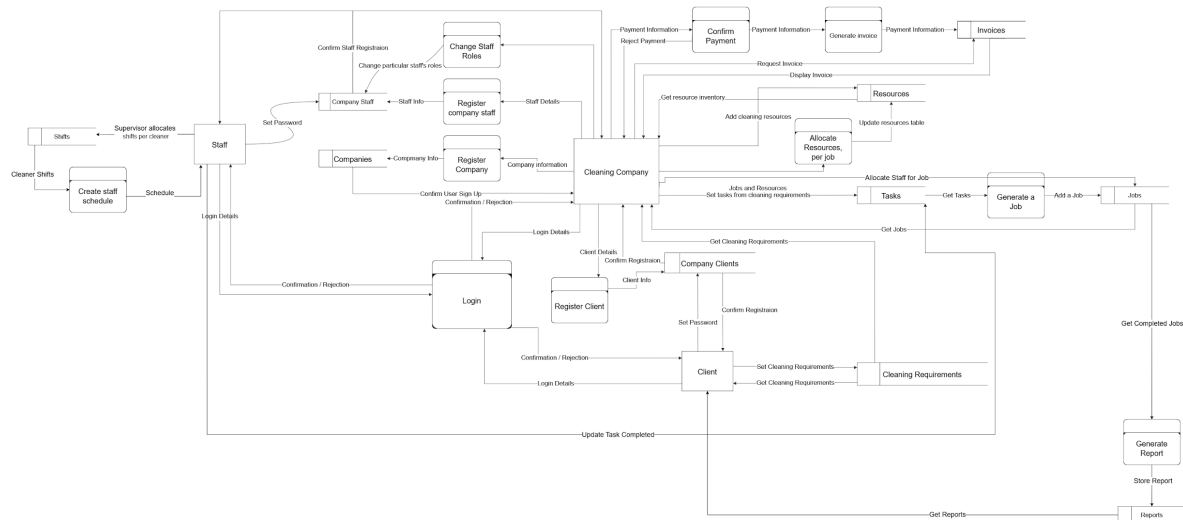


Figure 9. T50_Data_Flow_Diagram.png

4.0 Non-Functional Requirements

Performance

- **Response Time:** 90% of actions should be processed within 2 seconds.
- **Load Handling:** The system should support up to 200 concurrent users.

Reliability

- **Uptime:** The system should maintain 99.5% availability.
- **Error Recovery:** In case of errors, the system should allow recovery within 5 minutes without data loss.

Security

- **Data Security:** Encrypt all sensitive user data if any. (Clean maestro will handle the least amount of sensible data)
- **Role-Based Access:** Each user role should have designated access permissions.

Maintainability

- **Modular Code:** The system should follow modular design principles to enable easy updates. An MVC or microservice architecture approach will be taken into consideration.

- **Documentation:** Detailed system documentation should be maintained for future reference.

Portability

- **Compatibility:** The web app should support Chrome, Firefox, and Safari. The mobile app should be compatible with Android and, in the future, iOS.

5.0 Logical Database Requirements

Clean Maestro will utilize a relational database for data integrity and structure.

- **Data Format:** Standardized formats for client, task, and resource data.
- **Data Retention:** Retain report data for five years for audit and client review purposes.
- **Data Integrity:** Ensure real-time data synchronization across users and devices.

6.0 Other Requirements

User Training: Provide onboarding tutorials and training materials.

Technical Support: Offer technical support for system users.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Project Manager	Samuel Gallego Rivera	SGR	November 6, 2024.
Lead Architect	Jose Britto Saaji	JBS	November 6, 2024.
Frontend Developer	Dilvir Singh	DS	November 6, 2024.
Backend Developer	Akorede Daniel Osunkoya	ADO	November 6, 2024.