



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Faculty of
Computing

UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING, UTMJB

SEMESTER 2, SESSION 2024/2025

PROJECT PHASE 1

SECD2613 SYSTEM ANALYSIS AND DESIGN

SECTION 05

LECTURER'S NAME: TS. DR. MUHAMMAD IQBAL TARIQ BIN IDRIS

BY THE GROUP: GROUP 6

NAME	MATRIC NO
VIKASH RAJ A/L KATHIRASAN	A24CS0207
DASHITA A/P GAUTHAMAN	A24CS0065
MUHAMMAD ANAS BIN HAMDAN	A24CS0275
MUHAMMAD ARIF BIN MUHAMAD SUHAIMI	A24CS0126
GAN YAO TONG	A24CS0075
MOHAMMAD YAZID BIN MOHD KHAIRUDDIN	A24CS0117

Content

1.0	Overview of the Project.....	3
2.0	Problem Statement.....	4
3.0	Proposed Solutions.....	5
4.0	Current Business Process/Workflow.....	7
5.0	Logical DFD (AS-IS).....	9
6.0	System Analysis and Specification.....	15
6.1	Logical DFD TO-BE system.....	15
6.2	Process Specification.....	21
7.0	Physical System Design.....	24
7.1	Physical DFD TO-BE system.....	24
8.0	System Wireframe.....	45
9.0	Summary of the proposed system.....	49

1.0 Overview of the Project

ARS Builders Sdn. Bhd. is a reputable road construction company led by Mr. Sithamparanathan Arunasalam, specializing in the development and maintenance of road infrastructure projects. Currently, the company executes many of its construction projects through predominantly manual processes, which has resulted in significant operational issues. These include frequent human errors due to reliance on paper-based documentation, missed deadlines caused by poor time management, inefficient allocation and utilization of resources, and considerable difficulties in tracking and managing project-related data effectively.

As a consequence of these challenges, the company's overall operational efficiency is hampered, project delivery timelines are often compromised, and costs escalate beyond initial budgets, ultimately leading to decreased client trust and satisfaction. These systemic problems not only hinder the company's ability to grow but also damage its reputation in a competitive industry.

To tackle these inefficiencies and streamline its operations, ARS Builders Sdn. Bhd. is committed to enhancing the planning, coordination, and management of its projects through the implementation of an integrated project management solution. This strategic improvement aims to enable the company to deliver projects on schedule, maintain tighter control over budgets, and foster greater confidence and satisfaction among its clients. A thorough feasibility study has been conducted, which confirms the technical feasibility, economic justification, and operational suitability of the proposed solution for the company's needs.

2.0 Problem Statement

Before designing a new system, the problems faced by the current project management practices at ARS Builders Sdn. Bhd. must be first identified. Through interviews and document analysis, several notable operational challenges have emerged, primarily due to the company's continued reliance on manual planning and coordination. These issues have led to inefficiencies, project delays, cost overruns, and dissatisfaction among clients. By identifying these problems, it becomes clear why ARS Builders Sdn. Bhd. requires a more effective solution to improve project execution, resource utilization, and client relationships.

The problem statements are as follows:

- **Unrealistic Time Estimates Cause Missed Deadlines:** Project schedules are often planned with unrealistic timelines, resulting in projects not being completed on time. This frustrates clients and disrupts overall project workflows.
- **Budgets Frequently Exceed Estimates:** Poor planning and unexpected issues cause project costs to go beyond initial budgets, which affects profitability and increases stakeholder concerns.
- **Delays Lead to Client Dissatisfaction:** Repeated project delays have caused clients to lose confidence in the company's ability to deliver on promises, impacting future opportunities.
- **Inadequate Time and Budget Planning:** Management struggles to create accurate schedules and budgets, leading to confusion among staff and poor allocation of resources.
- **Difficulty Managing Unexpected Problems:** When unforeseen issues arise, the company lacks clear processes to respond effectively, resulting in further delays and increased costs.
- **Poor Coordination and Communication:** There is no centralized system to coordinate between personnel, equipment, and resources, leading to scheduling conflicts and inefficient workflows.
- **Lack of Clear Project Progress Tracking:** Without proper monitoring tools, it is challenging for management to gain accurate, real-time insights into project status, hindering oversight and timely decision-making.

3.0 Proposed Solutions

To overcome the problems currently faced by ARS Builders Sdn. Bhd., we proposed to make a desktop application that is specifically built to address all of their issues. Rather than relying on companies outside of ARS Builders to create the development and use all the tools and hardware, this desktop application will change the way that construction project delivery is done.

Feasibility Study:

The aim of this feasibility study is to evaluate the practicality of developing a desktop application for a construction project delivery. The company currently relies heavily on manual methods, which are prone to errors and inefficiencies. This new system seeks to streamline operations and improve scalability.

1. Technical Feasibility

Technology Stack: The system can be built using widely available technologies such as:

- Frontend (GUI): Python (Tkinter/PyQt), Java (JavaFX), or [Electron.js](#).
- Backend/Logic: Python, Java, or C#.
- Database: MySQL or SQLite (to be hosted locally).

Availability of Skills: All programming/ database technologies outlined to create the system are well documented and used widely by developers, which means that developing in-house or contracting all or part of it is realistic.

Infrastructure: The development of the system would need internet access and basic hardware (PC or laptop) for the admin dashboard. No specialized infrastructure is required.

Conclusion: Technically Feasible

2. Economic Feasibility

Estimated Costs:

- Development Costs: RM 50,000 (one-time)
- On-going Operational Costs: RM 12,000 (annually in maintenance or upgrades)

Expected Benefits:

- Time saved on planning and reporting
- Lower cost overruns because issues can be identified earlier
- Better labour/ equipment utilization
- Improved levels of client satisfaction → improved probabilities of receiving more awards of contract

Return on Investment (ROI): As we have operational savings and increased amount of project awards, Investment can be recovered in 3 to 4 years.

Conclusion: Economically feasible with positive ROI

3. Operational Feasibility

Ease of Use: User-friendly interface specifically designed for project manager, engineer, and admin access.

Training Requirements: Minimal training needed for staff to use the admin panel.

Customer Adoption: Expected to run smoothly with good management support as the technology has been built for internal use.

Conclusion: Operationally feasible

4.0 Current Business Process/Workflow

ARS Builders Sdn. Bhd. is utilizing a manual system, which is normal for most construction project management. Since there is no formal digital solution, all of these systems are not connected, causing operational inefficiency, duplicate work, and increased dissatisfaction from clients. The workflow is essentially portrayed as follows:

1) Project Planning/Scheduling

- Project Managers are gathering schedule information manually and mostly using a spreadsheet or hand-written notes.
- The time estimating is based solely on their experience (which results in unrealistic timelines).
- The managers found it completely unrealistic to update the information on the scheduling tool, especially if something unplanned occurred on site that caused disruption.

2) Resource Management

- All resource allocations of manpower, machinery, and materials are performed manually.
- The availability and use of resources across the sites are sometimes tracked on a white board, other times in a multitude of files scattered in different locations.
- This has resulted in clashes in resources, double bookings, or lapses in periods of inactivity.

3) Communication/Coordination

- The exchange of information is tracked or communicated via WhatsApp, phone calls, and emails.
- There is no formal method for tracking heavily on decisions, updated status from site, and assigning responsibilities on platform.
- The gaps in communication lead to miscommunication, repeat actions, and delays in taking action.

4) Progress Tracking

- Progress on site is typically reported verbally, or through informal reports.
- Stakeholders typically do not have a formal methodology of reporting their progress, or tracking key performance indicators (KPI's), as they occur.
- Management does not have sight of where the project is actually at a given time, relative to where it is supposed to be.

5) Budgeting & Cost Tracking

- Budget estimates and costs were done in a series of disconnected spreadsheets.
- The actual costs to date are reconciled only at the end of the project.
- Factors such as overtime or weather delays that contributed to unplanned costs are not considered in the base estimates.

6) Permit & Compliance Tracking

- Permits were both applied for and tracked manually.
- The system does not incorporate alerts for permit deadlines/requirements.
- Reviews typically encounter delays due to missing approvals/submissions.

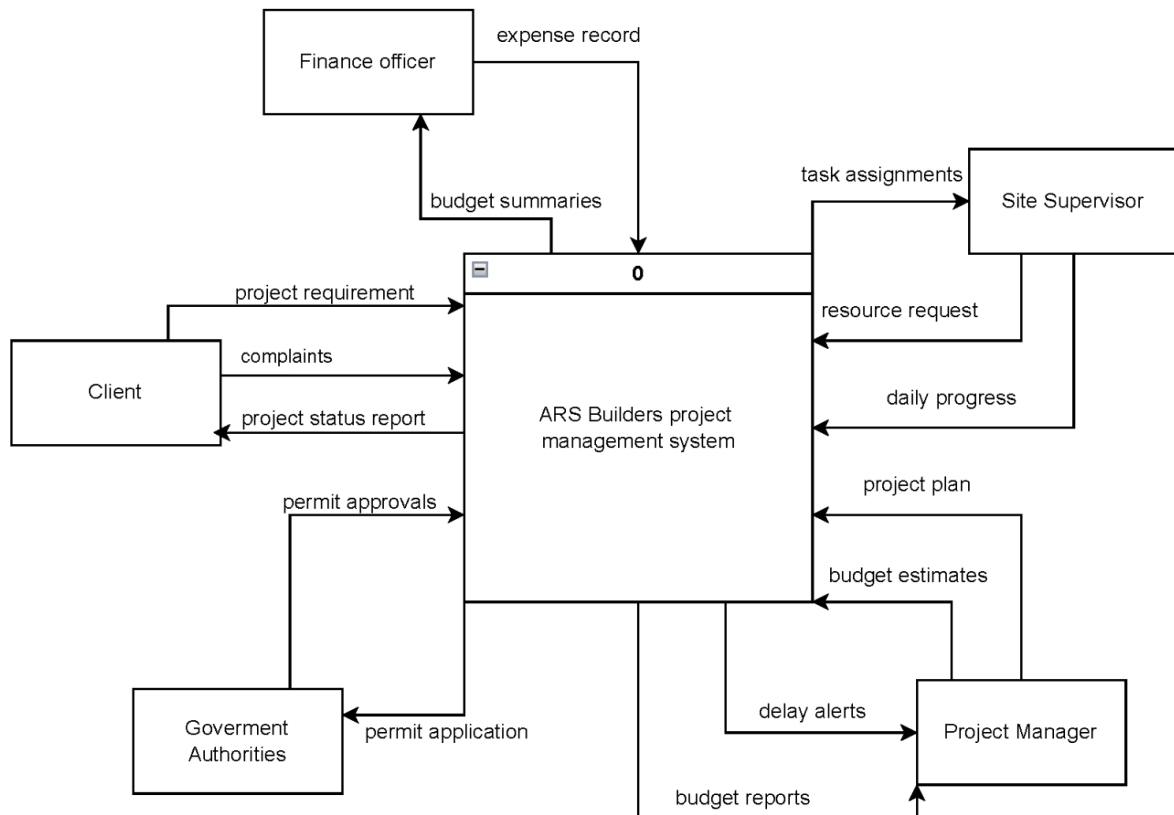
Operational Issues Summary

- A high potential for human error with manual entry.
- Poor visibility/real-time updates.
- Poor coordinates on project tasking/improvement.
- Client dissatisfaction due to delays, overruns, and breakdown in communications.

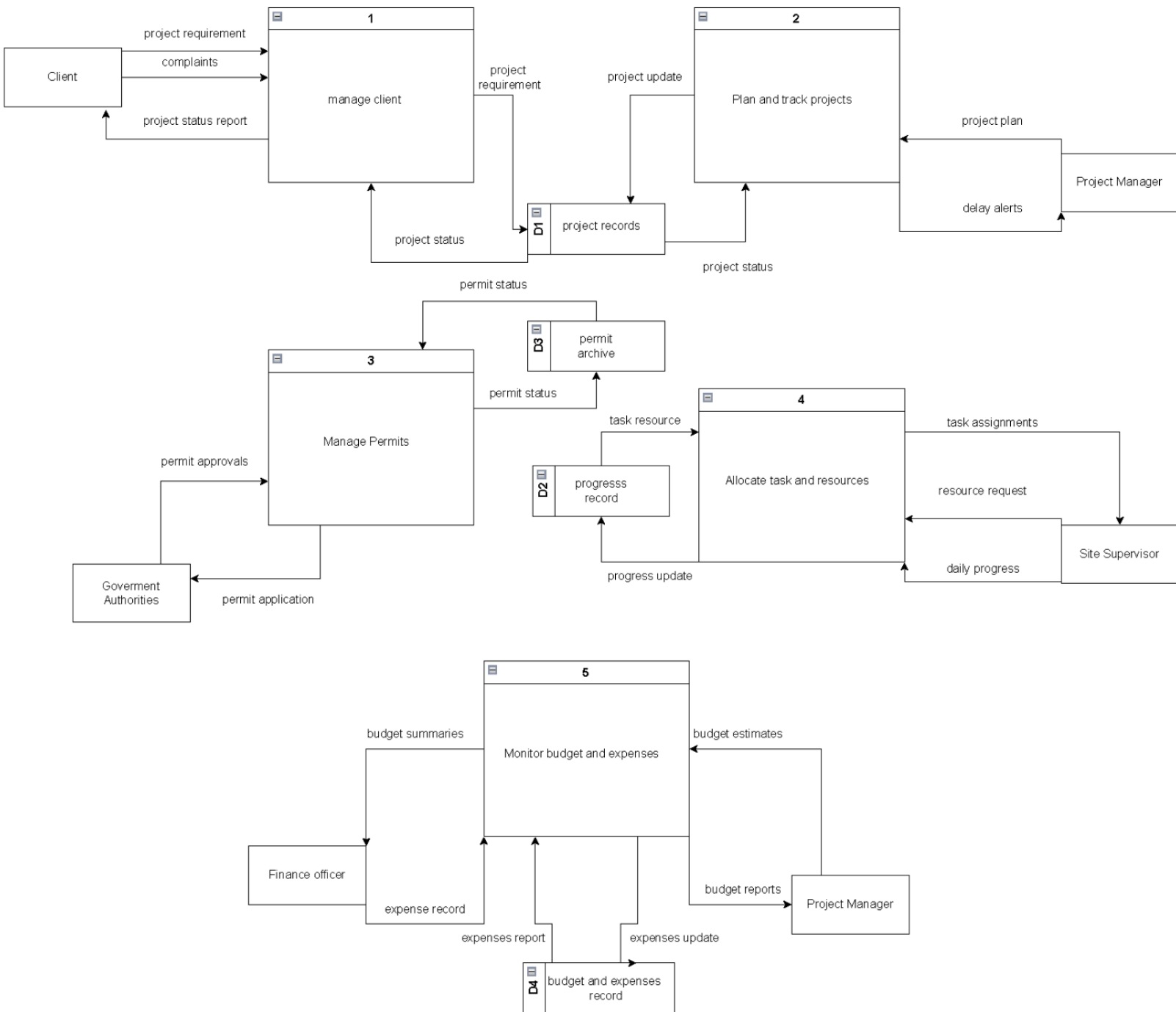
5.0 Logical DFD (AS-IS)

To visualise the proposed system, we had created a logical data flow diagram that showed the proposed process of the system. In this logical DFD, we had modified the AS-IS DFD to automate some of the processes that can be automated, while ensuring the project management process is maintained.

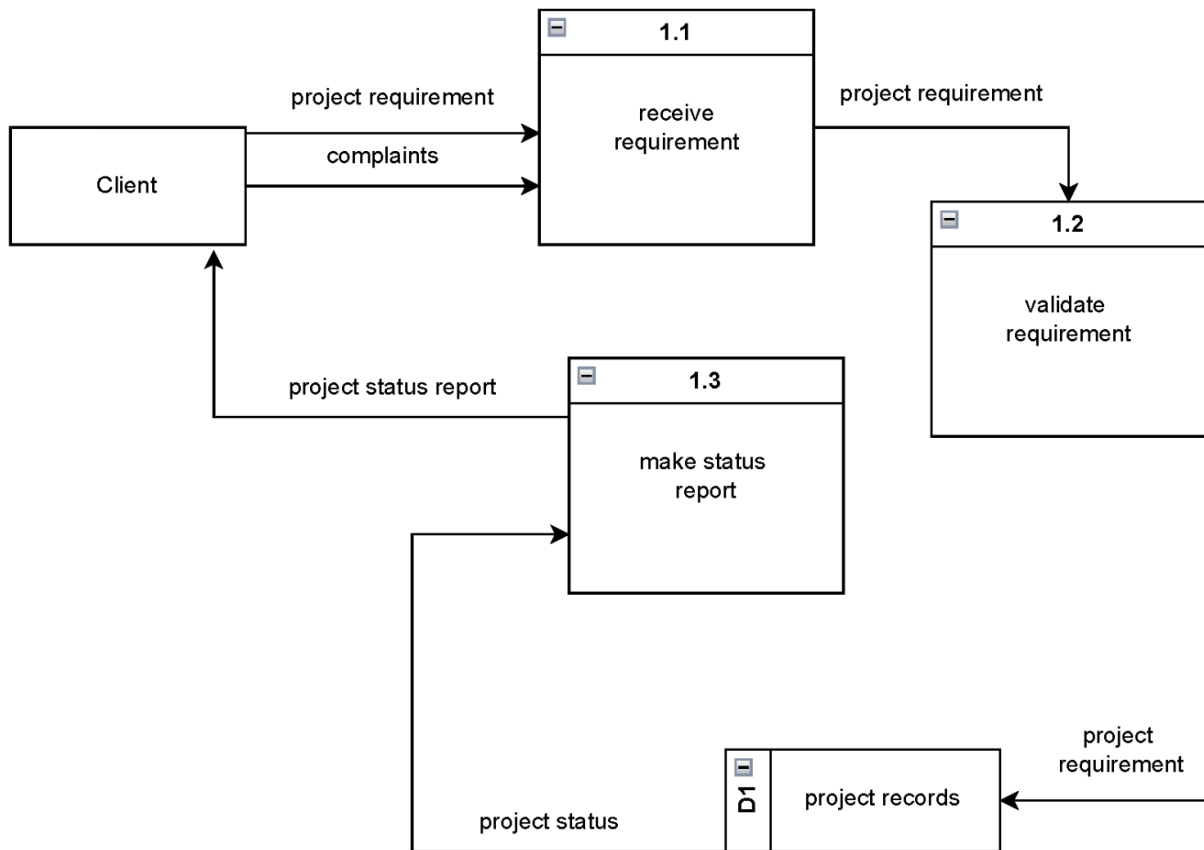
Context Diagram



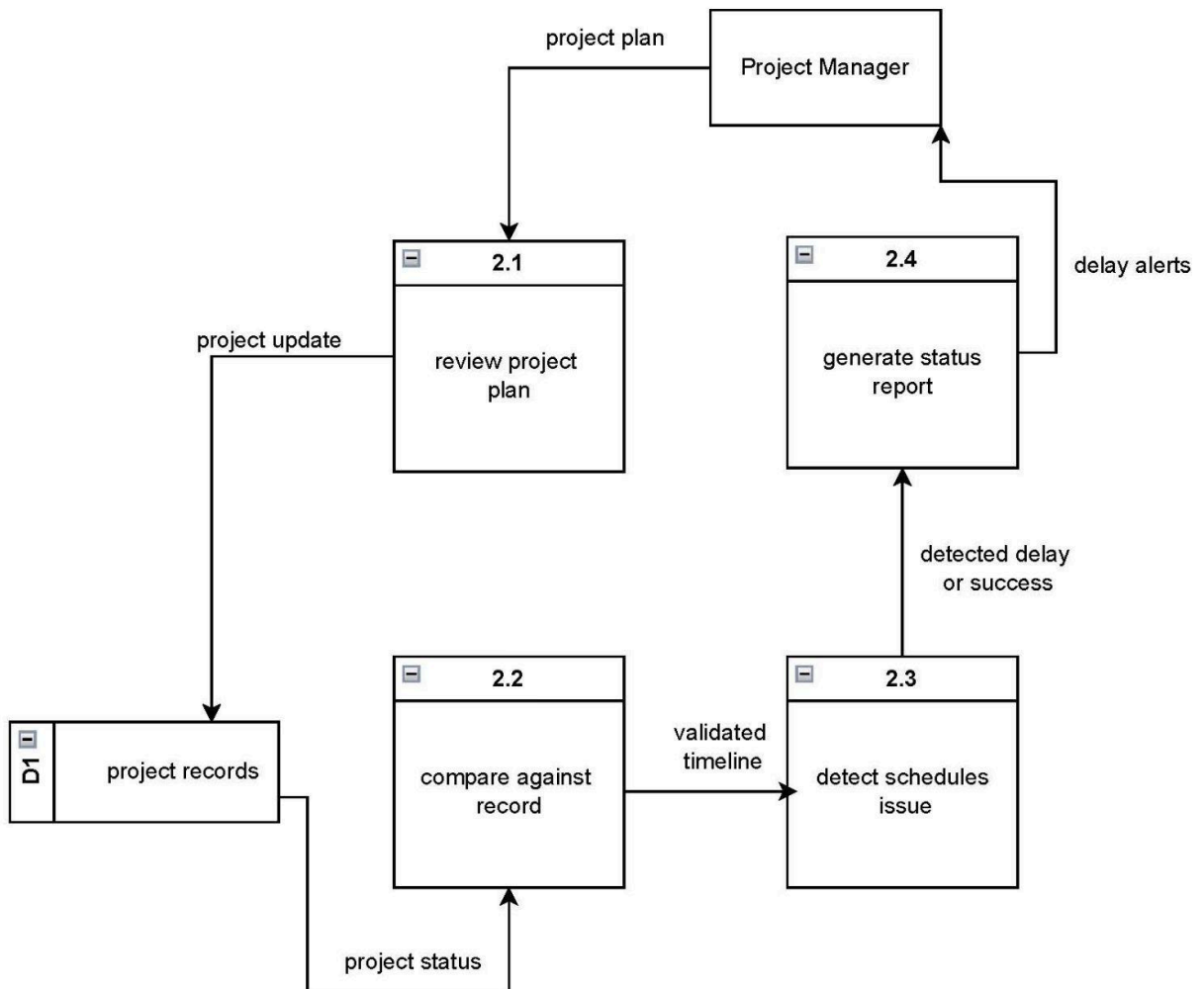
Level 0 DFD



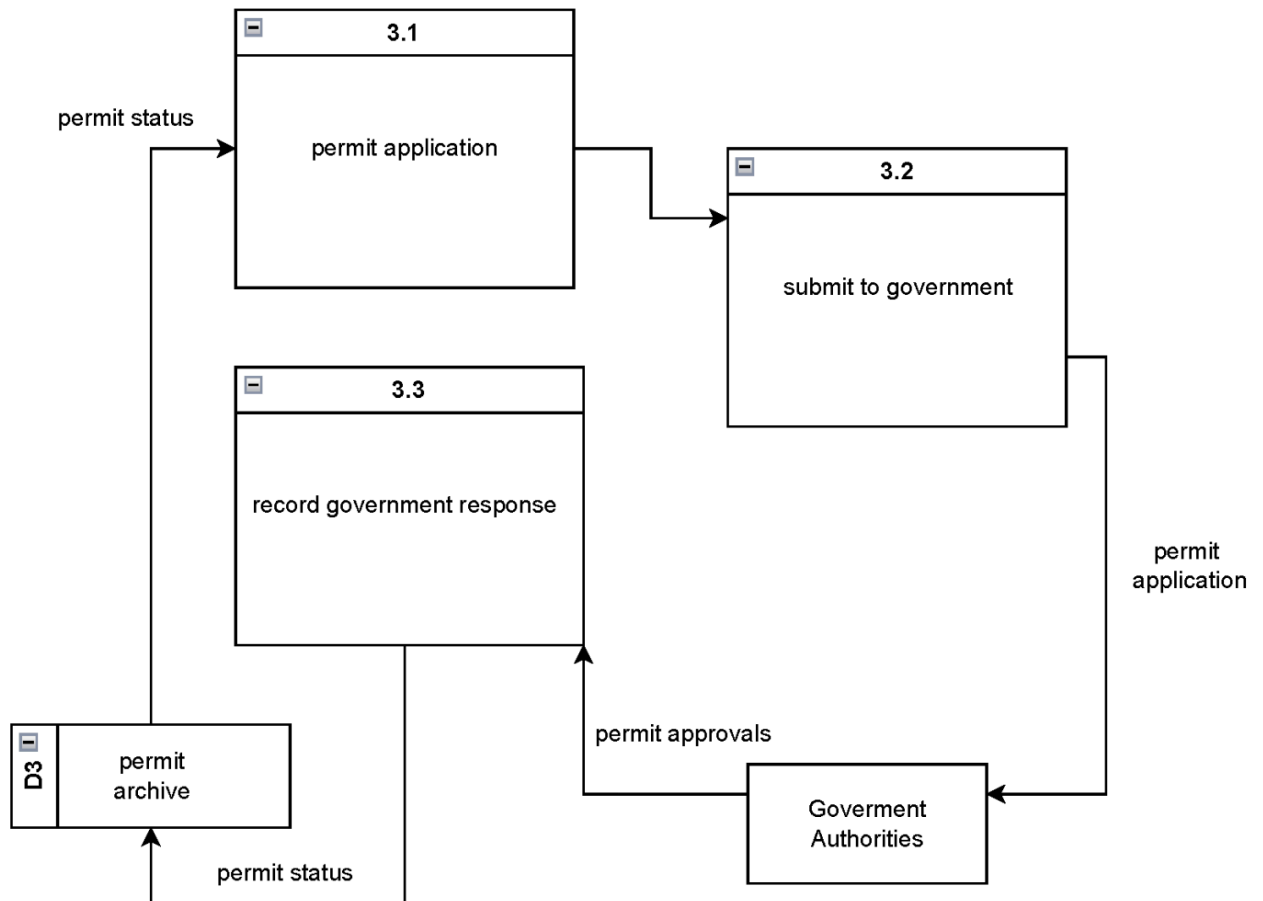
Child Diagram of Process 1



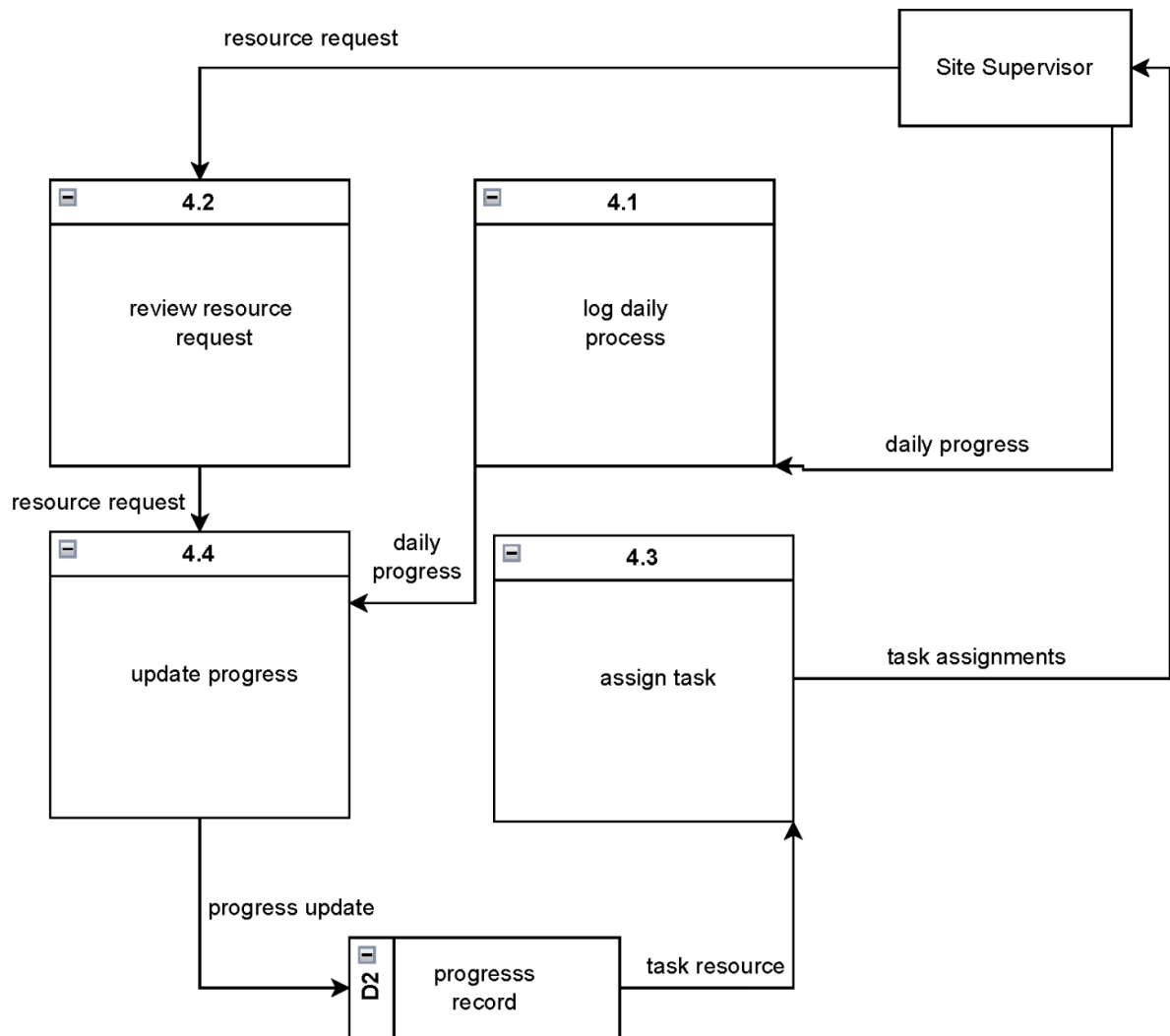
Child Diagram of Process 2



Child Diagram of Process 3



Child Diagram of Process 4

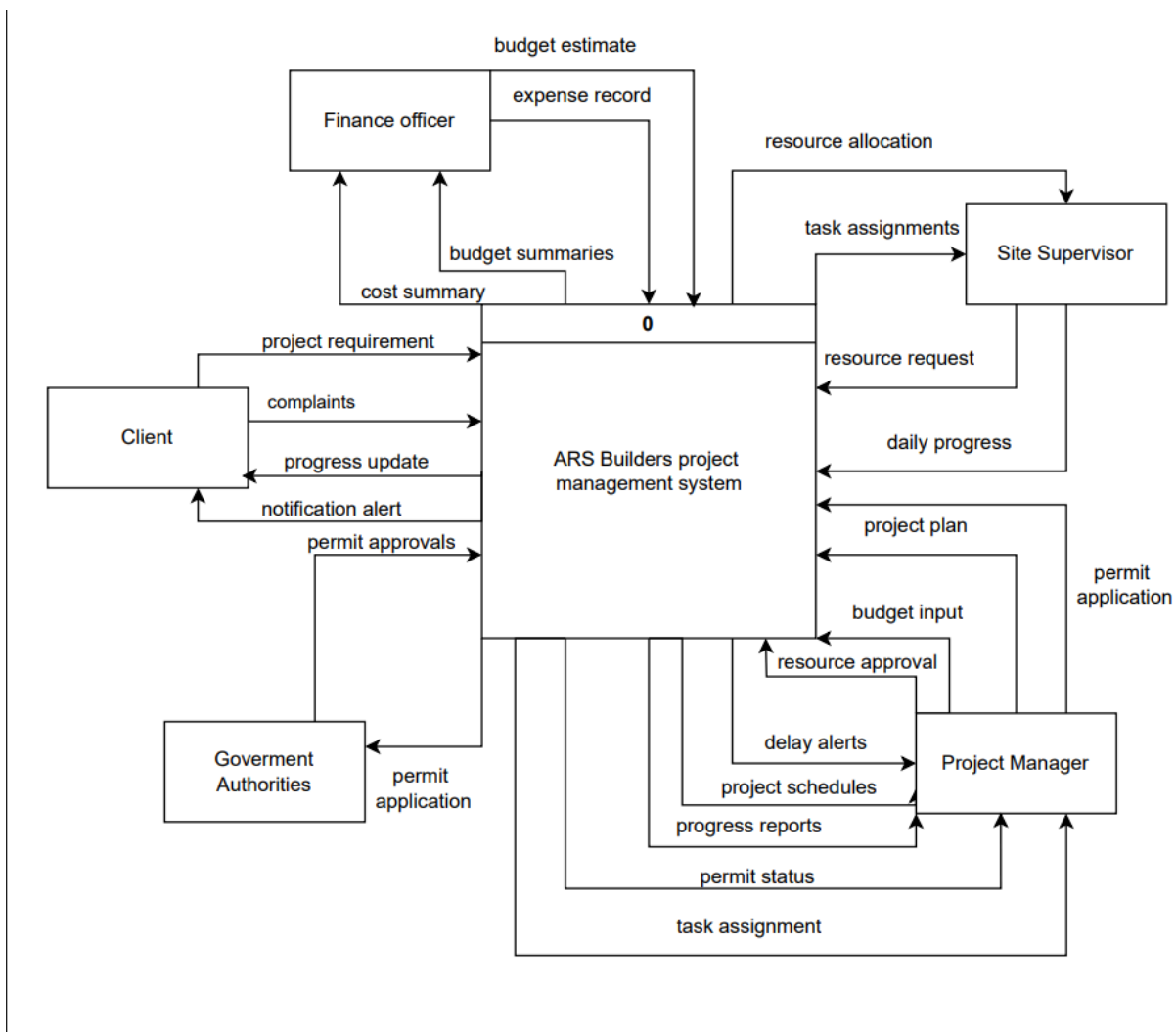


6.0 System Analysis and Specification

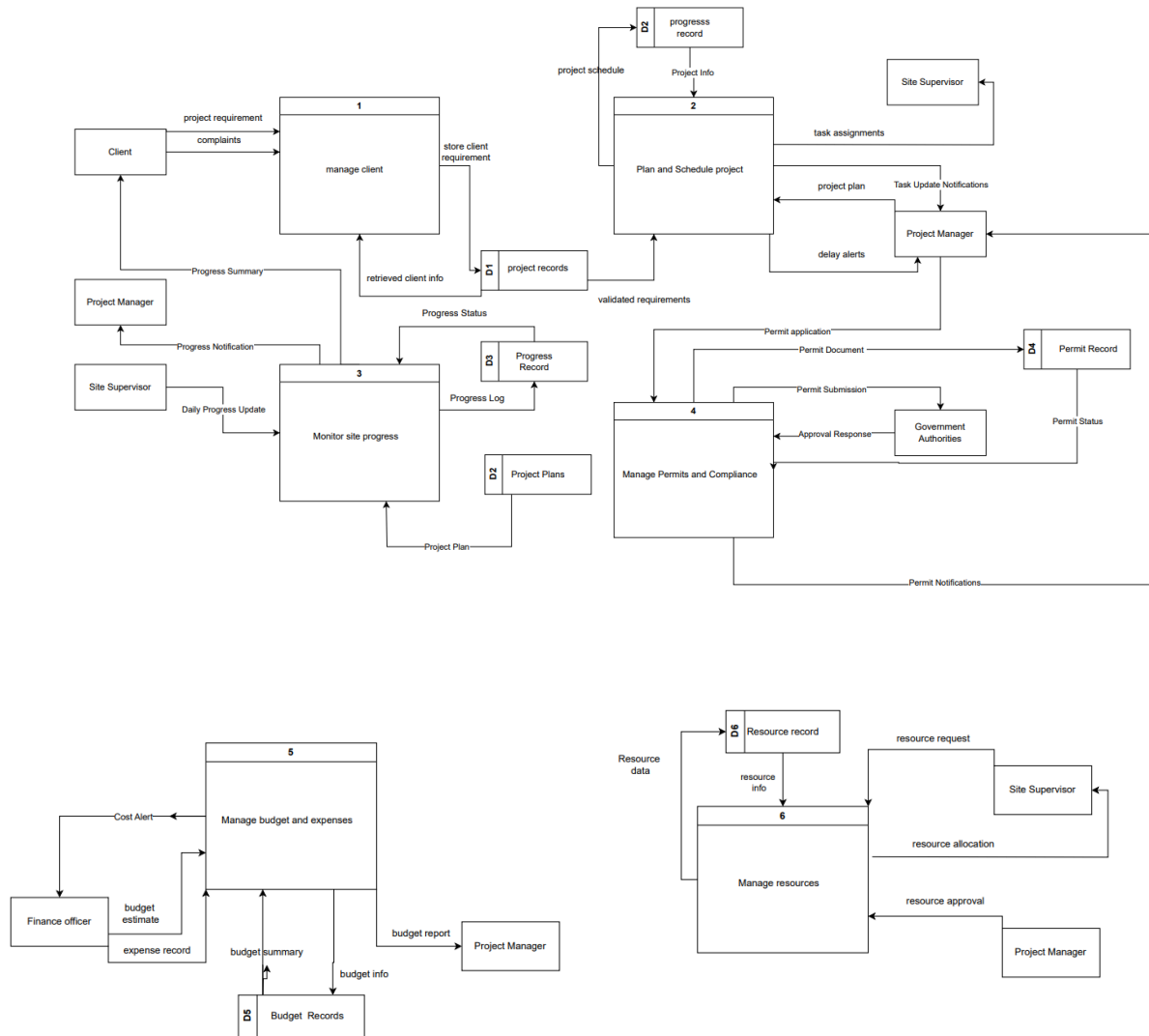
Based on stakeholder needs, data flow requirements, and process specifications, this section describes the improved functional structure of the suggested ARS Builders Project Management System. This section's goal is to describe the TO-BE system's logical architecture in detail and demonstrate how it resolves the operational issues that have been identified.

6.1 Logical DFD TO-BE system

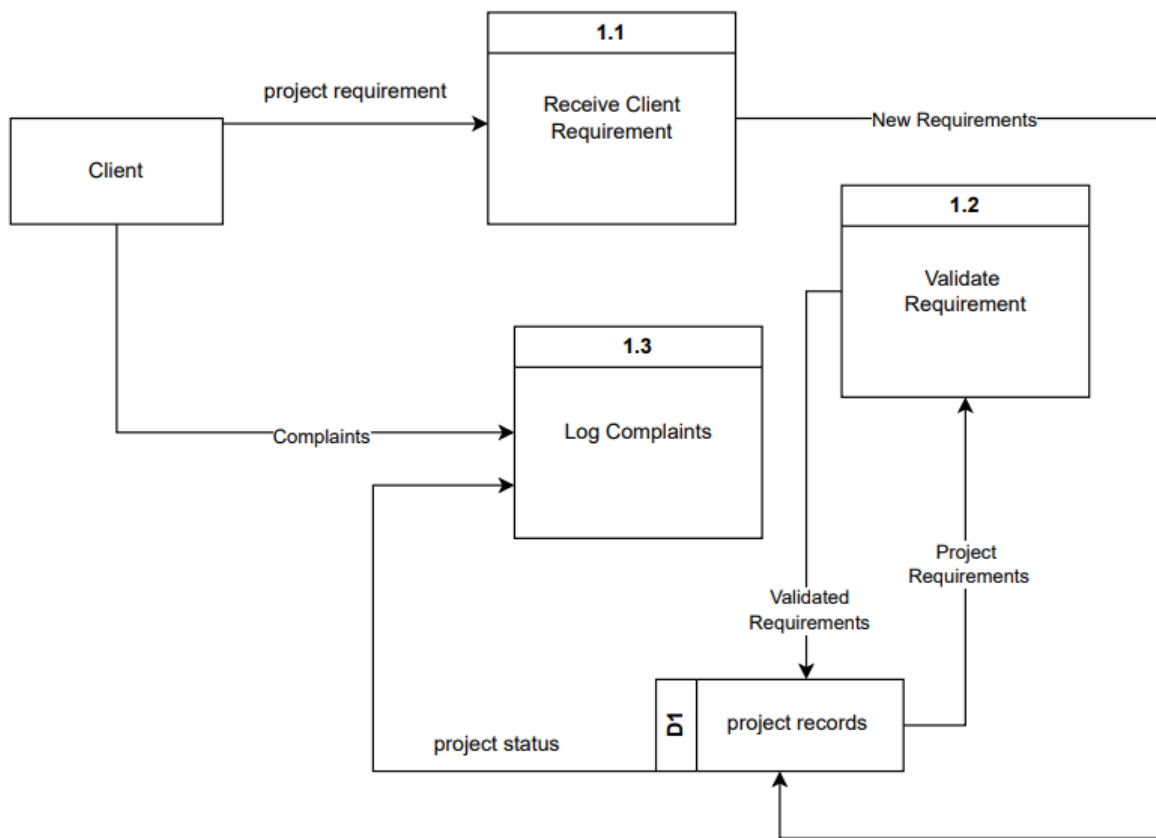
Context Diagram



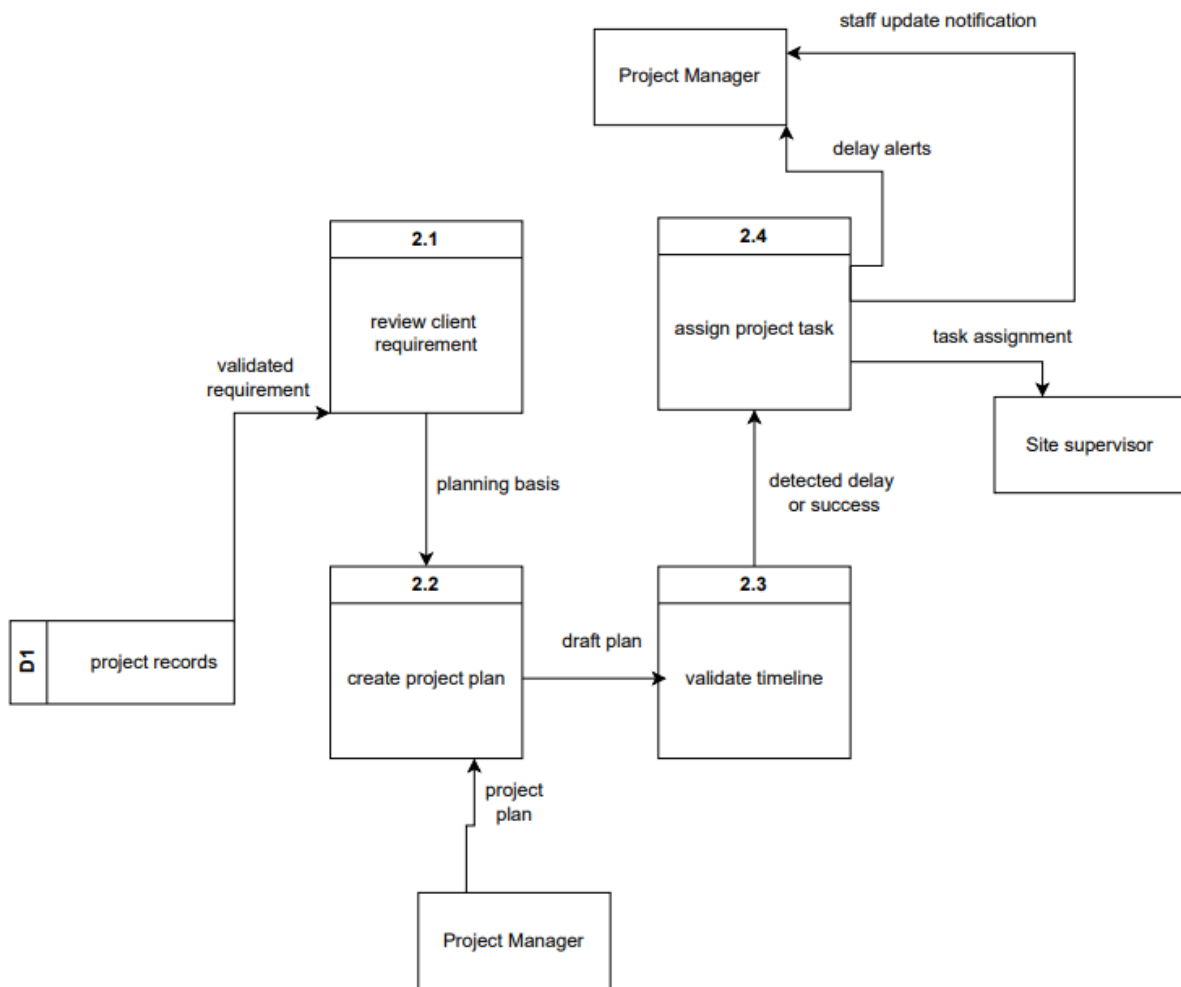
Level 0 diagram



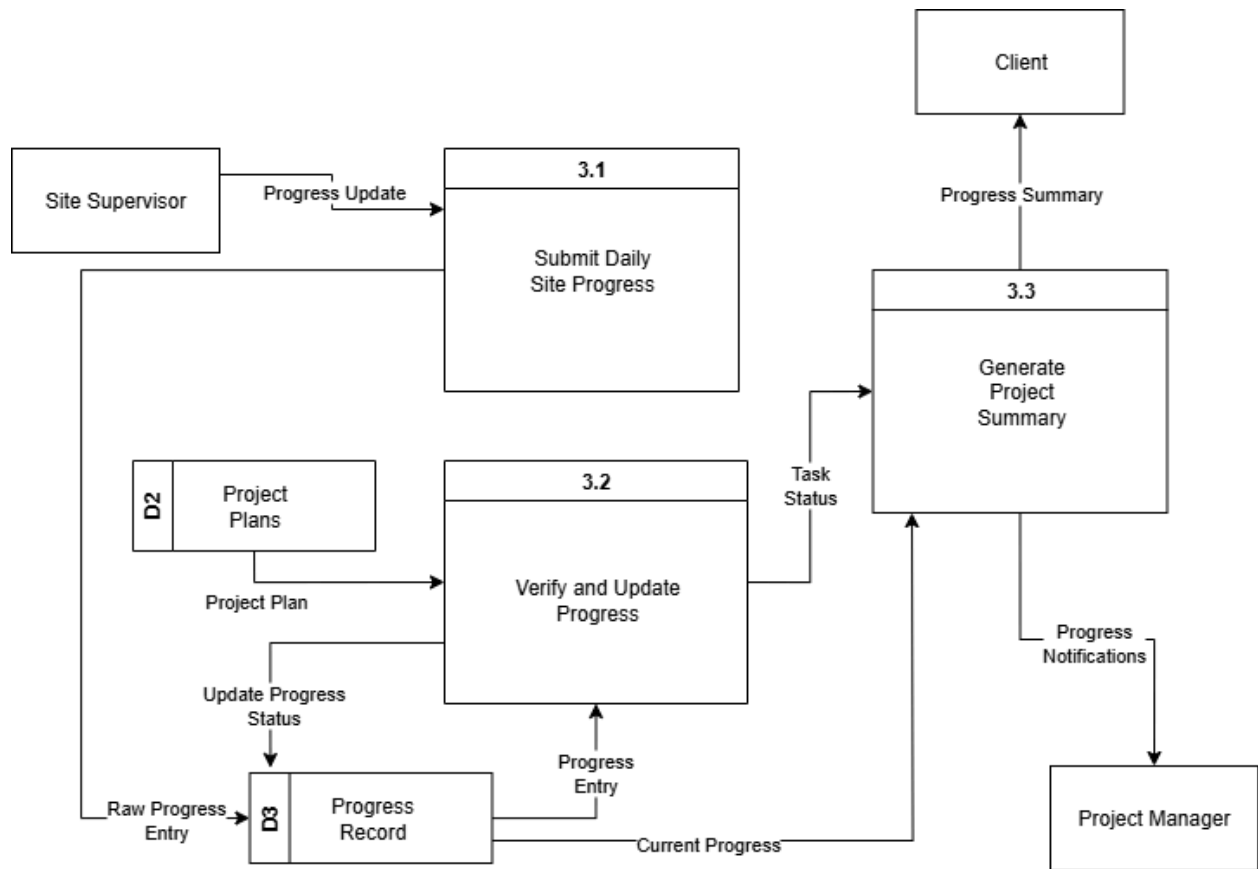
Child 1 diagram



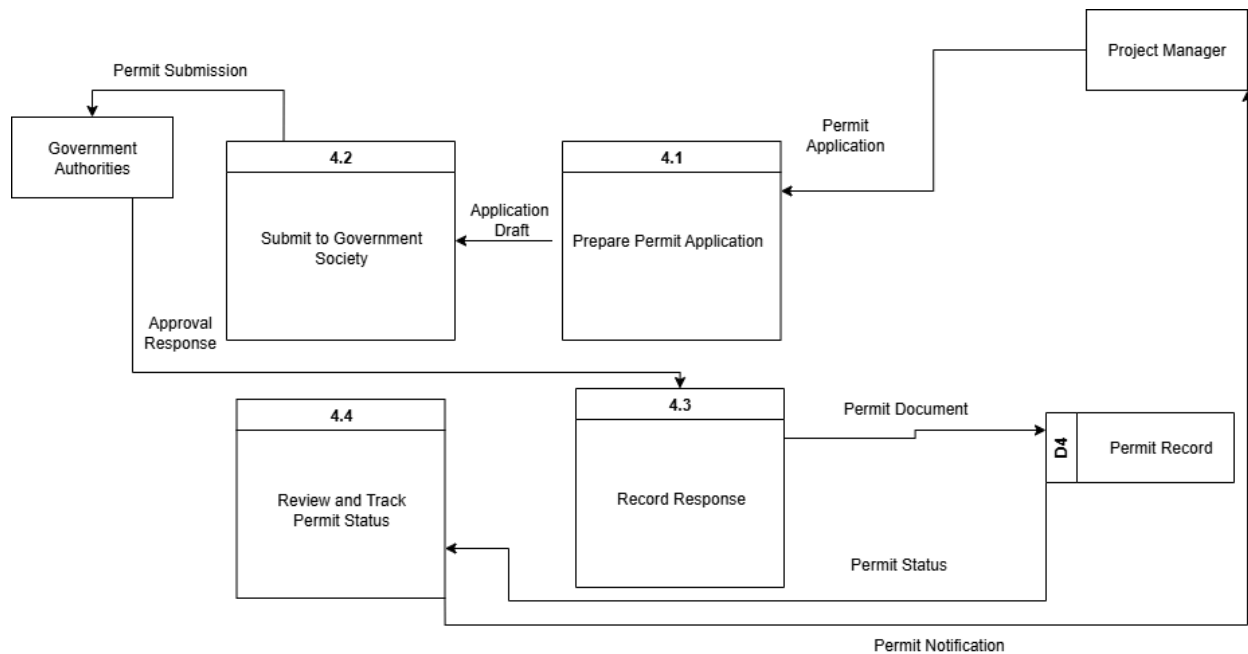
Child 2 diagram



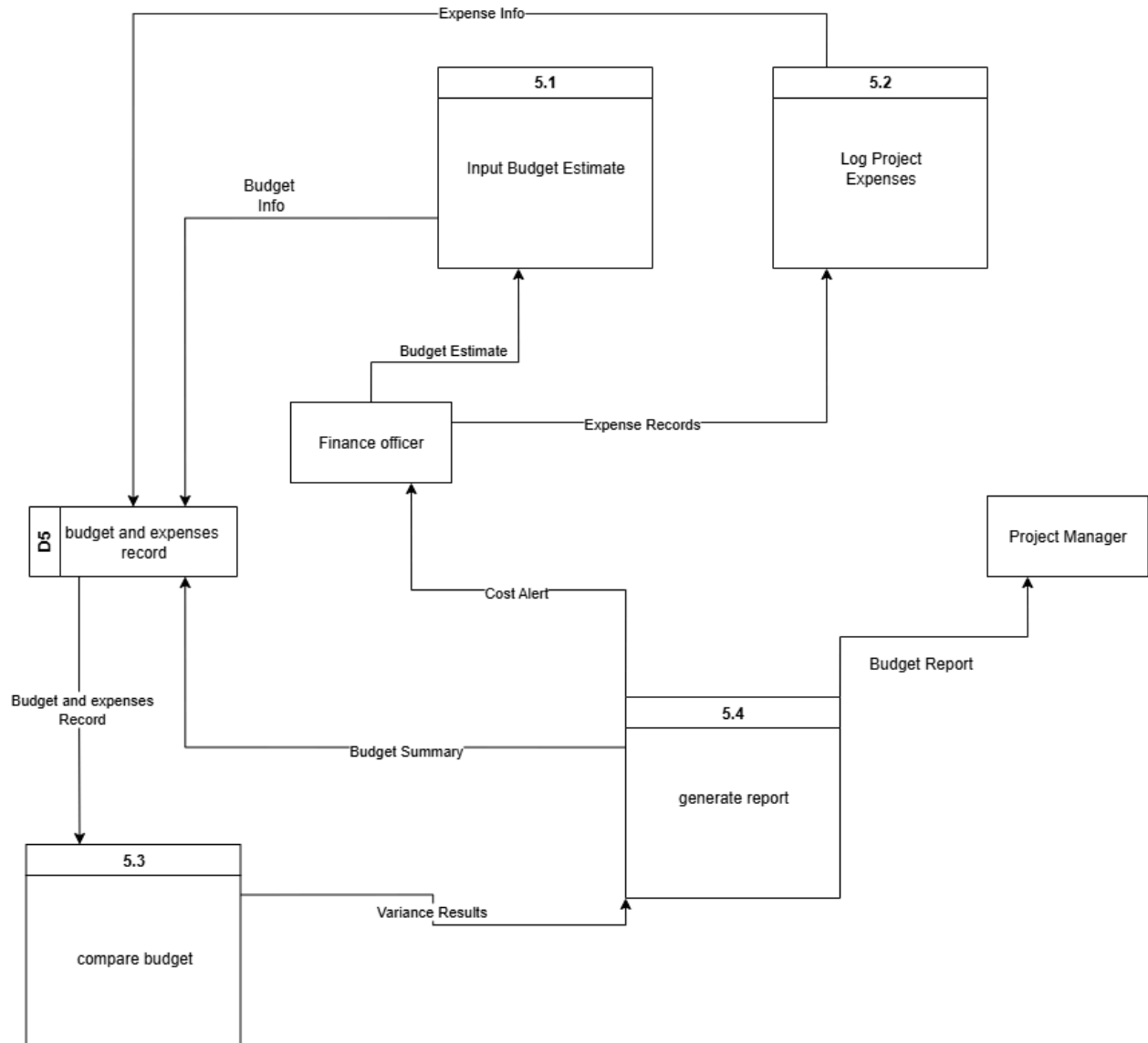
Child 3 diagram



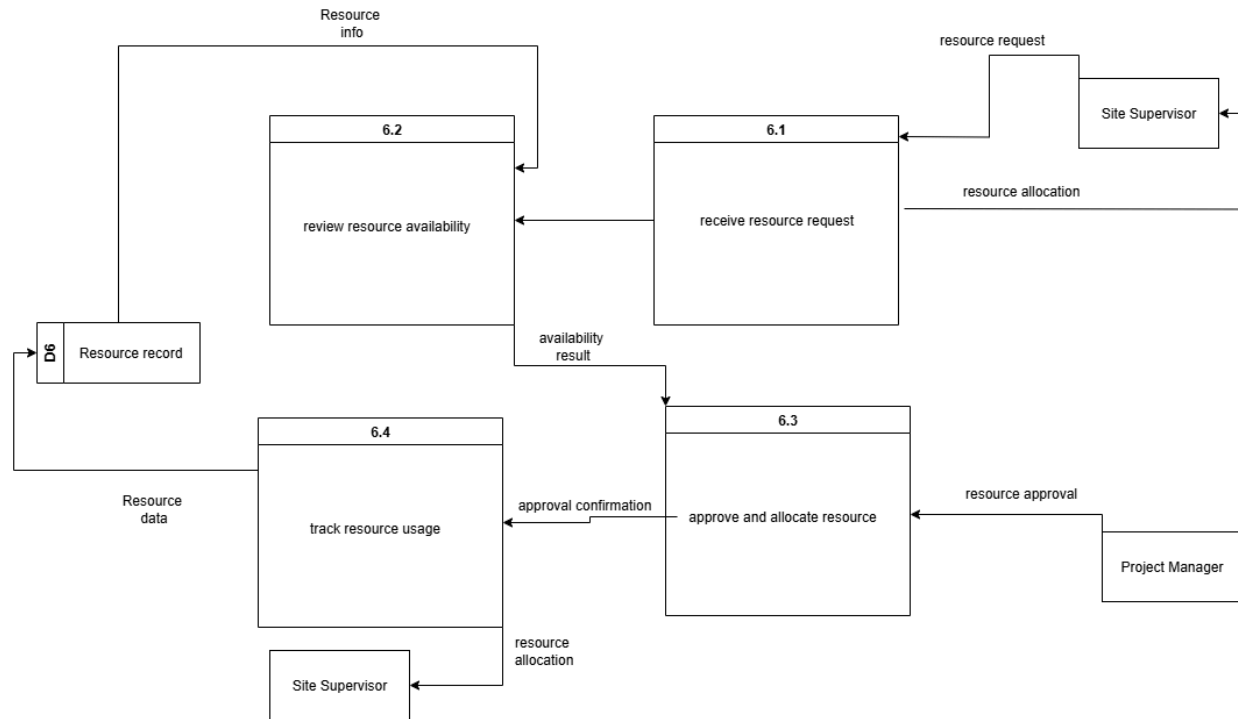
Child 4 diagram



Child 5 diagram



Child 6 diagram



6.2 Process Specification

Process 1: Manage Client Request

Handles submission and validation of project requirements and complaints from clients. Ensures data is stored for later planning and provides feedback to clients and project managers.

Sub-processes:

- 1.1 Receive Client Requirement – Receives new project requirements from the client and stores them.
- 1.2 Validate Requirement – Validates and updates the requirement in the client records.
- 1.3 Log Complaint / Feedback – Stores complaints or feedback submitted by the client
- 1.4 Notify Project Manager – Forwards validated requirements and complaint alerts to the Project Manager.

Process 2: Plan & Schedule Project

Enables the Project Manager to plan tasks and timelines based on validated client requirements, detect schedule conflicts, and assign tasks to the Site Supervisor.

Sub-processes:

- 2.1 Review Client Requirement – Reads client requirements to begin planning.
- 2.2 Create Project Plan – Combines requirement input with project manager's planning to generate a project plan.
- 2.3 Validate Timeline and Dependencies – Checks the new plan against existing schedules for conflicts.
- 2.4 Assign Project Tasks – Assigns tasks to Site Supervisor and notifies Project Manager

Process 3: Monitor Site Progress

Captures daily updates from the Site Supervisor, compares them with the project plan, stores progress records, and notifies both the Project Manager and Client of project status.

Sub-processes:

- 3.1 Submit Daily Site Progress – Stores raw progress input from the Site Supervisor.
- 3.2 Verify and Update Progress – Compares progress with the project plan and updates KPIs.
- 3.3 Generate Project Summary – Summarizes progress for the Project Manager and Client.
- 3.4 Alert on Delay or Issue – Identifies delays or issues and alerts both PM and Supervisor.

Process 4: Manage Permits & Compliance

Manages preparation, submission, and tracking of permits. Sends applications to government authorities and notifies the Project Manager of updates.

Sub-processes:

- 4.1 Prepare Permit Application – Generates permit application from PM input.
- 4.2 Submit to Government Authority – Sends application to external authorities.
- 4.3 Record Government Response – Stores approval or rejection from the authority.
- 4.4 Review and Track Permit Status – Enables PM to check and track permit status.

Process 5: Manage Budget & Expenses

Receives budget estimates and expense records, compares them, and generates variance reports and cost alerts for both Finance Officer and Project Manager.

Sub-processes:

- 5.1 Input Budget Estimate – Logs initial estimates from Finance Officer.
- 5.2 Log Project Expenses – Captures expenses as they occur.
- 5.3 Compare Budget vs Actual – Evaluates discrepancies and triggers alerts if needed.
- 5.4 Generate Budget Report – Creates summary reports for PM and Finance.

Process 6: Manage Resources

Handles resource requests from the Site Supervisor, checks availability, and updates allocation records after PM approval. Sends assignment instructions to the supervisor.

Sub-processes:

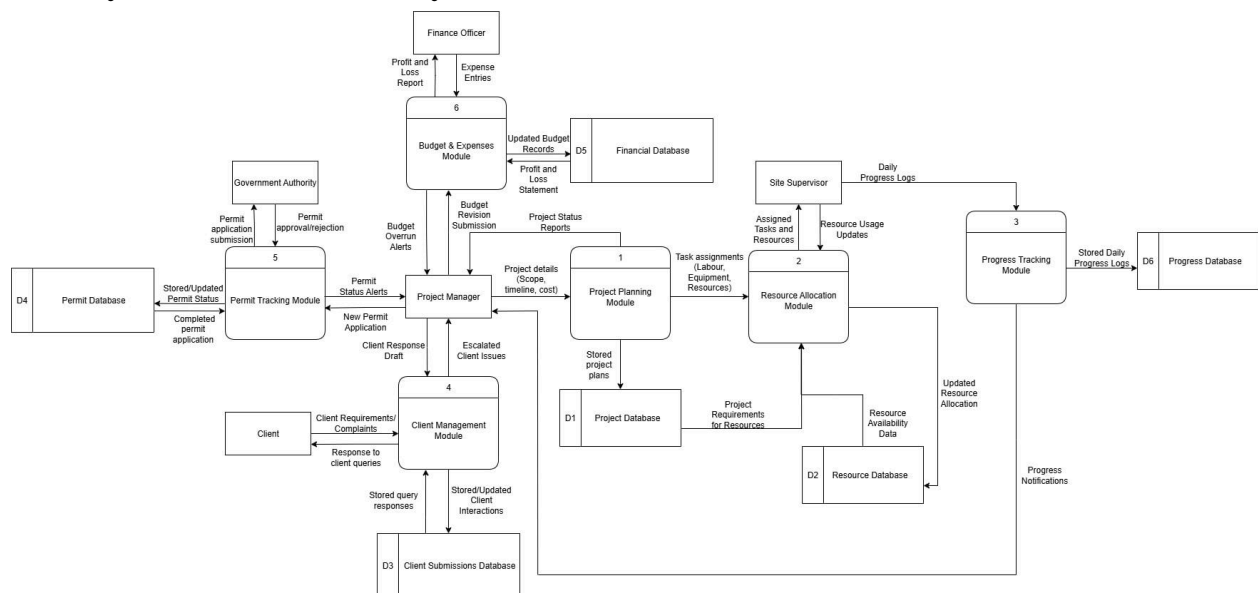
- 6.1 Receive Resource Request – Logs request into resource database.
- 6.2 Check Resource Availability – Reads current resource data.
- 6.3 Approve Resource Request – Uses PM approval to proceed.
- 6.4 Assign Resource – Updates the resource record and sends task assignment.

D6 – Resource Records (write: request & allocation; read: for availability check and

7.0 Physical System Design

After analysing and specifying the requirement and function of the new system, we have designed the system by identifying how the system may carry those requirements and functions, whether by computerised method or still through manual method. Other than that, we also identified the type of computer or device that will be used for the system and also the system architecture to ensure the workings of the system when it is implemented.

7.1 Physical DFD TO-BE system



This system streamlines the lifecycle of a construction project—from planning and resource allocation, to tracking progress, handling client interactions, managing permits, and overseeing budget and expenses. The system comprises six interconnected modules and six major data stores.

1. Project Planning Module

- Input: Project details (scope, timeline, cost) from the Project Manager.
- Output: Task assignments (labour, equipment, resources) to Resource Allocation Module.
- Stores project plans in **Project Database (D1)**.

2. Resource Allocation Module

- Inputs:
 - Task assignments from Project Planning Module.
 - Resource availability data from **Resource Database (D2)**.
- Interacts with **Site Supervisor** by sending task/resource assignments and receiving usage updates.
- Retrieves **project resource requirements** from the **Project Database**

3. Progress Tracking Module

- Input: Resource allocation and usage updates from Resource Allocation Module and Site Supervisor.
- Stores daily logs in **Progress Database (D6)**.
- Sends progress notifications to Project Manager.

4. Client Management Module

- Handles client complaints/queries.
- Sends response drafts to Project Manager.
- Stores client interactions in **Client Submissions Database (D3)**.

5. Permit Tracking Module

- Submits permit applications to **Government Authority** and tracks approvals/rejections.

- Sends status alerts to Project Manager.
- Stores permit status in **Permit Database (D4)**.

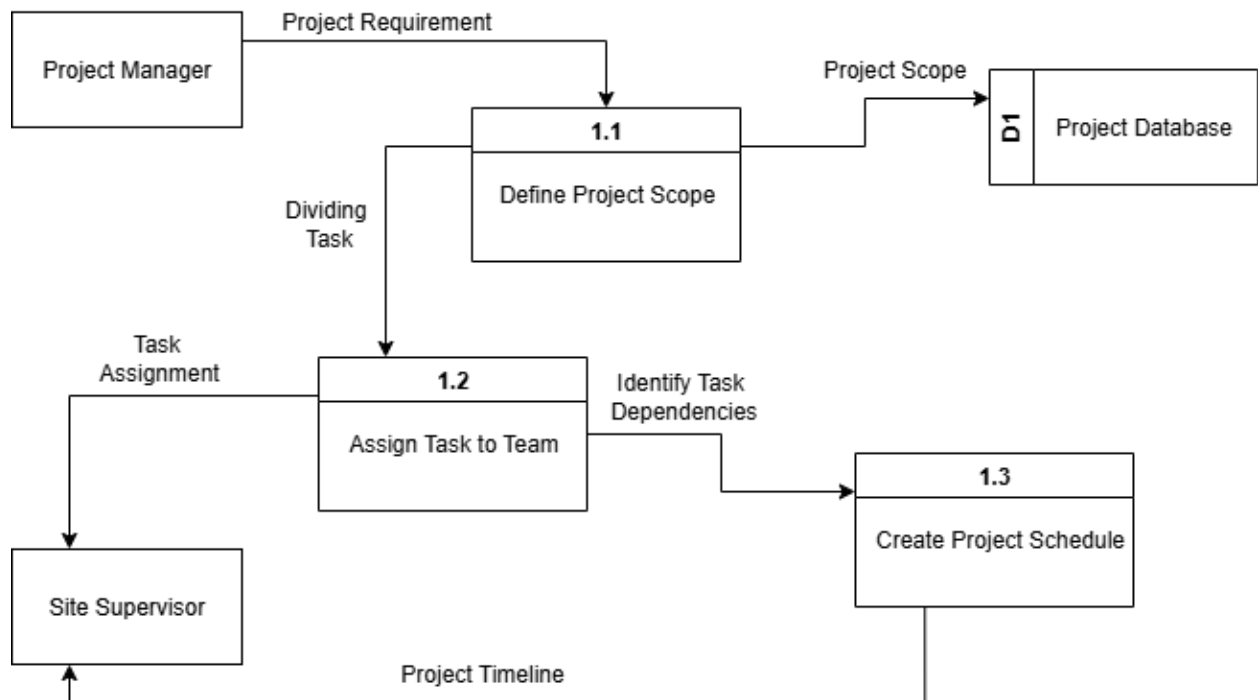
6. Budget & Expenses Module

- Receives expense entries from **Finance Officer**.
- Generates profit and loss reports/statements.
- Sends budget overrun alerts and updates to Project Manager.
- Updates **Financial Database (D5)**.

Entities

- **Project Manager**: The **central controller** of the system; coordinates all modules including planning, resource allocation, client management, permit tracking, progress tracking, and budgeting. Receives and sends critical information such as project status reports, budget revisions, and permit status alerts.
- **Site Supervisor**: Handles **on-site activities**. Receives assigned tasks and resources from the system and provides **resource usage updates** and **daily progress logs** to support progress tracking.
- **Client**: Submits **requirements or complaints** and receives responses. May escalate issues through the Client Management Module to the Project Manager.
- **Government Authority**: Processes **permit applications** submitted through the Permit Tracking Module and sends back **approvals or rejections**.
- **Finance Officer**: Manages **financial records**, enters expenses, and generates **profit and loss reports**. Interacts with the Budget & Expenses Module to ensure updated financial tracking.

Child 1 diagram



This diagram is a **Child DFD** that elaborates on the "**Plan Project**" process from the parent (Level 0) Data Flow Diagram. It shows a detailed breakdown of the planning activities required to define the project scope, assign tasks, and create a schedule. This process ensures that the project progresses from initial requirements to actionable, time-bound tasks assigned to the right team members.

Processes

Process 1.1 – Define Project Scope

This process begins when the **Project Manager** provides the **Project Requirement**. The system analyzes the requirement and formulates a detailed **Project Scope**.

- The defined scope is then stored in the **Project Database (D1)** for documentation and future reference.
- Output: **Project Scope**

Process 1.2 – Assign Task to Team

Based on the scope defined in 1.1, this process **divides the project into tasks** and assigns them to the appropriate teams or individuals.

- These task assignments are directed to the **Site Supervisor**, who will manage the team on the ground.
- The process also **identifies task dependencies**, which are critical for effective scheduling in the next process.
- Output: **Task Assignment, Task Dependencies**

Process 1.3 – Create Project Schedule

Using the task dependencies identified in 1.2, this process generates the **Project Timeline**.

- It sequences tasks based on logical and time constraints, and forwards the finalized timeline to the **Site Supervisor** to coordinate implementation.

External Entities

- **Project Manager**: Initiates the planning process by submitting the **Project Requirement**.

- **Site Supervisor:** Receives **task assignments** and **project timeline** to execute and manage daily operations on-site.
-

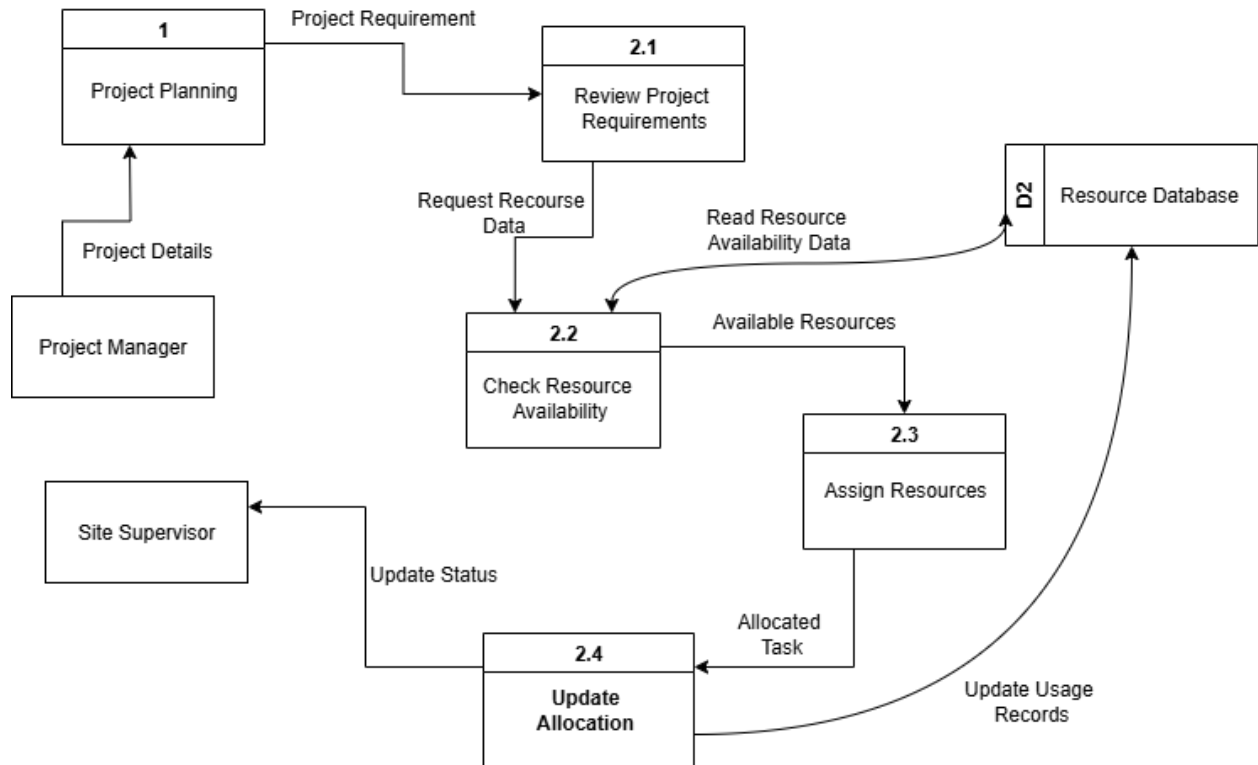
Data Store

- **D1 – Project Database:** This stores the **Project Scope** for use in future processes and auditing.
-

Data Flows

- **Project Requirement** flows from the Project Manager to Process 1.1.
- **Project Scope** flows to the Database (D1) and to 1.2.
- **Task Assignment** and **Timeline** are sent to the Site Supervisor.
- **Task Dependencies** are passed to Process 1.3 for scheduling.

Child 2 diagram



This Child DFD expands on the **Resource Allocation** process from the parent (Level 0) diagram. It provides a detailed view of how the system handles the assignment of resources—such as manpower, materials, or equipment—based on project requirements. The goal is to ensure resources are efficiently allocated and tracked throughout the project lifecycle.

Processes

Process 2.1 – Review Project Requirements

- **Input:** Project Requirement from the parent process (Project Planning).

- **Function:** Reviews the specific needs for the project to determine what resources are needed.
 - **Output:** A **request for resource data** is sent to the next process.
-

Process 2.2 – Check Resource Availability

- **Input:** Resource data request from 2.1.
 - **Function:** Retrieves data from the **Resource Database (D2)** to determine which resources are available.
 - **Data Store Involved:**
 - **Reads** from **D2 (Resource Database)** to get availability info.
 - **Output:** Sends **available resources** data to the next process.
-

Process 2.3 – Assign Resources

- **Input:** Available resources from 2.2.
 - **Function:** Allocates appropriate resources to the specific tasks based on availability and requirement.
 - **Output:** Sends **allocated tasks** to Process 2.4 and **updates resource usage records** in **D2**.
-

Process 2.4 – Update Allocation

- **Input:** Allocated task data from 2.3.
- **Function:** Updates the status of resource allocation.

- **Output:**
 - Sends **updated allocation status** to the **Site Supervisor**.
 - Also updates **usage records** in the **Resource Database (D2)**.
-

External Entities

- **Project Manager:** Provides **project details** and initiates the resource allocation process.
 - **Site Supervisor:** Receives **status updates** about which resources have been assigned and are ready for deployment.
-

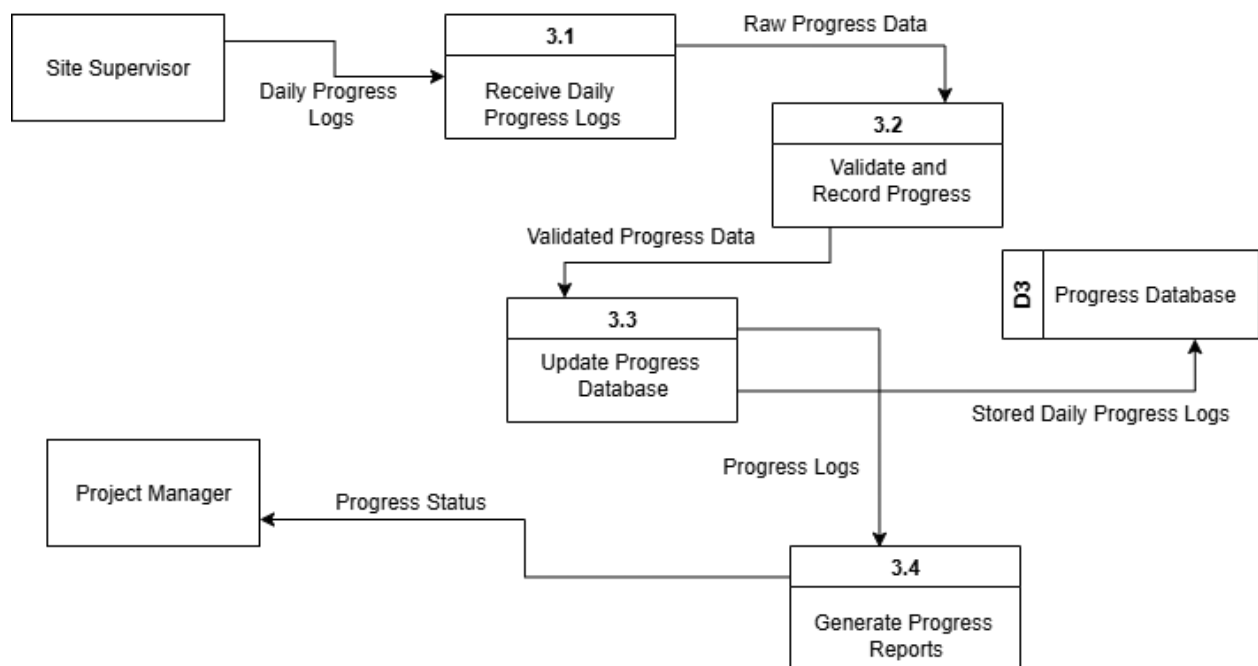
Data Store

- **D2 – Resource Database:** A centralized database that:
 - **Provides** real-time availability data.
 - **Stores** updated usage records for tracking and future planning.
-

Data Flows

- **Project Requirement** flows from the parent process into 2.1.
- **Resource data requests** are made from 2.2 to D2.
- **Available resources** flow to 2.3 for task allocation.
- **Allocated tasks** flow to 2.4.
- **Updates** are sent to D2 and the Site Supervisor.

Child 3 diagram



This Child DFD elaborates on the “**Monitor Project Progress**” function from the parent diagram. It describes how daily work updates from the project site are handled and transformed into structured, meaningful reports for project control and decision-making.

Processes

Process 3.1 – Receive Daily Progress Logs

- **Input:** **Daily Progress Logs** from the Site Supervisor.
 - **Function:** Captures raw daily progress data from the field.
 - **Output:** Forwards the **Raw Progress Data** to be validated in Process 3.2.
-

Process 3.2 – Validate and Record Progress

- **Input:** Raw data from 3.1.
 - **Function:** Ensures that progress logs are accurate and consistent. This prevents errors or fake reporting.
 - **Output:** Produces **Validated Progress Data**, which is sent to the database and next processes.
-

Process 3.3 – Update Progress Database

- **Input:** Validated progress data from 3.2.
 - **Function:** Updates the **Progress Database (D3)** with verified daily records.
 - **Output:** Stores **Daily Progress Logs** into D3 and makes the data available for reporting.
-

Process 3.4 – Generate Progress Reports

- **Input:** Progress logs retrieved from the database (via 3.3).
 - **Function:** Compiles and formats the data into **progress reports**.
 - **Output:** Delivers **Progress Status** to the **Project Manager**, providing insight into project health and timelines.
-

External Entities

- **Site Supervisor:** Supplies daily on-site progress updates.
 - **Project Manager:** Receives finalized progress reports to track project performance and make informed decisions.
-

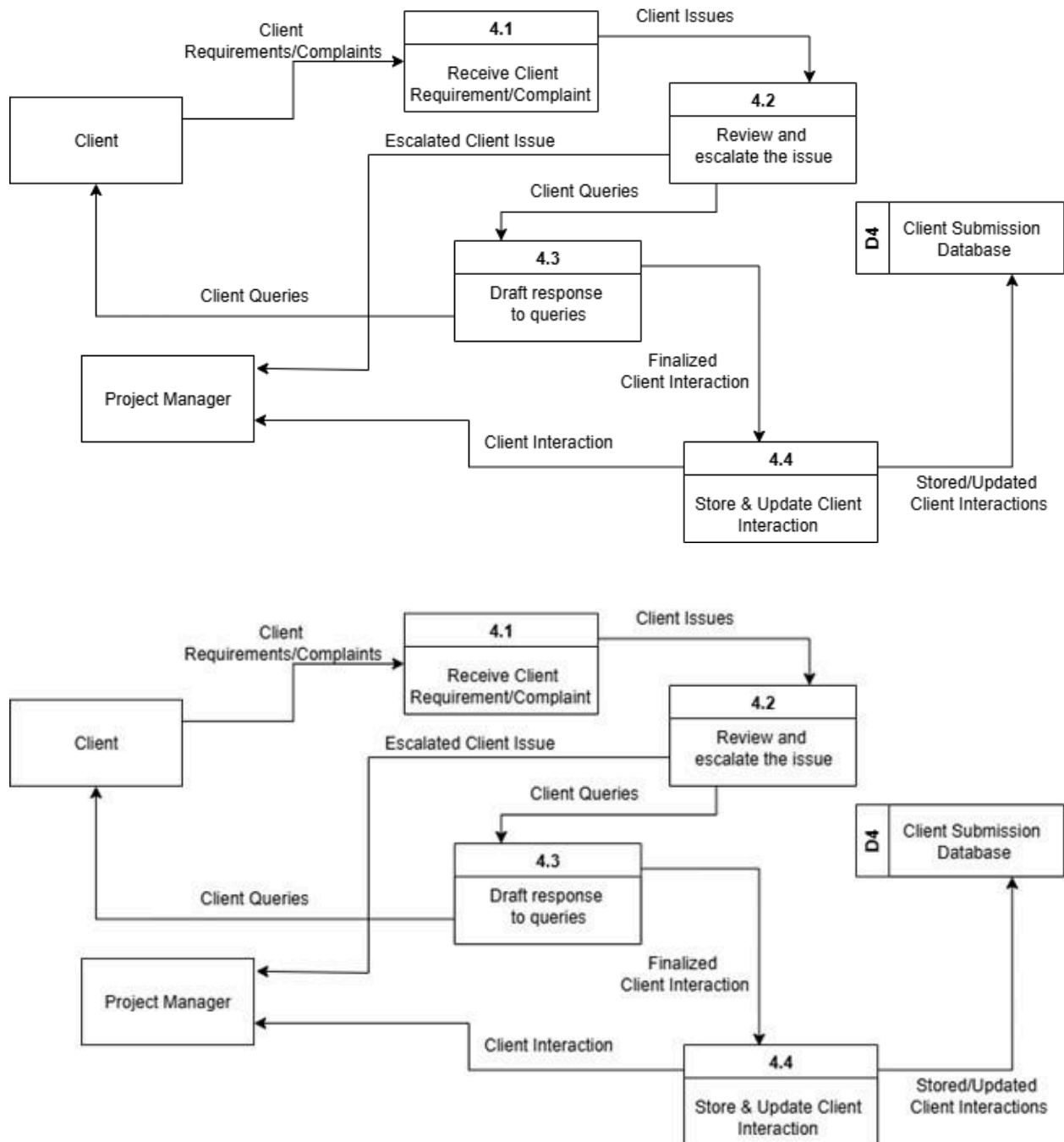
Data Store

- **D3 – Progress Database:** Stores all validated and updated progress logs for historical tracking and reporting purposes.
-

Data Flows

- **Daily Progress Logs** from the site supervisor are entered into the system at Process 3.1.
- **Validated Data** is recorded in the **Progress Database (D3)** via Process 3.3.
- **Progress Reports** are generated in 3.4 and sent to the Project Manager for decision-making.

Child 4 diagram



This diagram expands the "**Manage Client Communication**" process from the parent (Level 0) DFD. It focuses on how the system processes client requirements or complaints, handles escalations, provides responses, and maintains a record of client communications.

Processes Explained

Process 4.1 – Receive Client Requirement/Complaint

- **Input:** Client Requirements or Complaints.
 - **Function:** Accepts incoming messages from the client, whether they are service requests, inquiries, or complaints.
 - **Output:**
 - Forwards **Client Issues** to Process 4.2 if escalation is needed.
 - Forwards **Client Queries** to Process 4.3 for drafting a response.
-

Process 4.2 – Review and Escalate the Issue

- **Input:** Client Issues from 4.1.
 - **Function:** Assesses the complexity or severity of client complaints.
 - **Output:** Sends **Escalated Client Issues** back to **Project Manager** for resolution and **Client Queries** to 4.3
-

Process 4.3 – Draft Response to Queries

- **Inputs:** Client Queries from 4.2.
- **Function:** Prepares a suitable and informative response to the client.
- **Interactions:** Coordinates with the **Project Manager** if necessary to resolve issues.
- **Output:** Sends **Finalized Client Interaction** to 4.4 for storage, and to the **Client** for feedback.

Process 4.4 – Store and Update Client Interaction

- **Input:** Finalized client interaction from 4.3.
- **Function:** Records and updates the history of client communications for accountability and future reference.
- **Data Store Involved:**
 - **D4 – Client Submission Database**
 - Stores all **client interactions**, including complaints, queries, and responses.

External Entities

- **Client:** Sends queries or complaints and receives responses.
- **Project Manager:** Consulted when escalated issues or complex client questions need higher-level input or decision-making.

Data Store

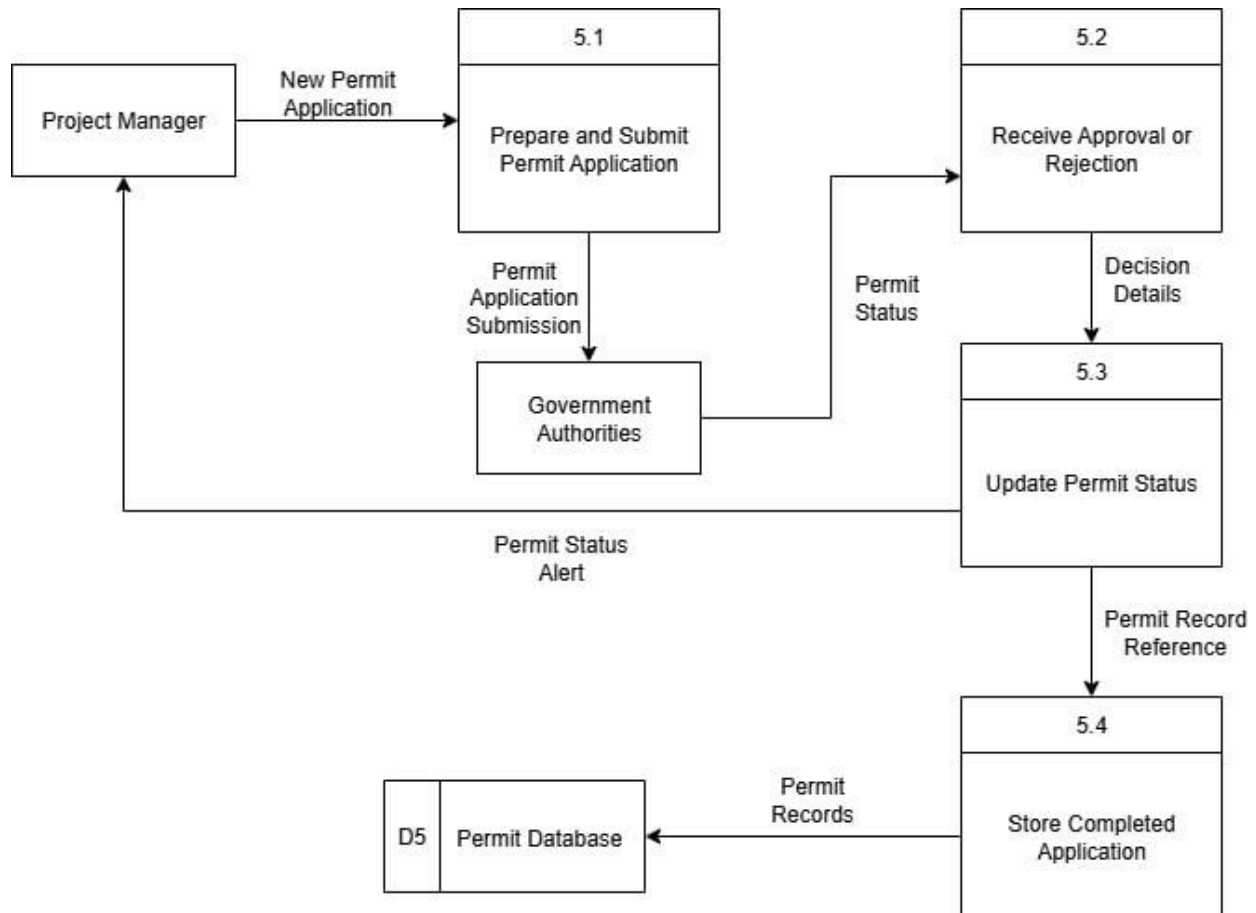
- **D4 – Client Submission Database:** Keeps all stored and updated client communications for historical tracking and auditing.

Data Flows

- **Client Requirements/Complaints** go into Process 4.1.
- **Escalated Issues** and **Client Queries** move through validation and response (4.2, 4.3).

- **Finalized Interaction** is delivered back to the **Client** and also **stored** in D4.

Child 5 diagram



This diagram expands on the “**Manage Permit Process**” function from the parent (Level 0) DFD. It shows the step-by-step data flow involved in applying for government permits needed for the execution of a project, covering submission, status tracking, and record-keeping.

Processes

Process 5.1 – Prepare and Submit Permit Application

- **Input:** New Permit Application from the **Project Manager**.
 - **Function:** Gathers necessary information and documentation to fill out the permit form.
 - **Output:** Sends the **Permit Application Submission** to the **Government Authorities** for approval.
-

Process 5.2 – Receive Approval or Rejection

- **Input:** Response from Government Authorities regarding the submitted application.
 - **Function:** Captures the **Permit Status** (Approved or Rejected).
 - **Output:** Sends **Decision Details** to Process 5.3 to update the status.
-

Process 5.3 – Update Permit Status

- **Input:** Decision details from 5.2.
 - **Function:** Updates the system's status based on government feedback.
 - **Output:** Notify the **Project Manager** with **Permit Status Alerts** and send **Permit Record Reference** to process 5.4
-

Process 5.4 – Store Completed Application

- **Input:** Permit record reference from 5.3.
- **Function:** Updates the system's status based on government feedback.
- **Data Store Involved:** Sends the **Permit Records** to **D5 – Permit Database**.
- **Output:** Stores the **Permit Records** in the **Permit Database**

External Entities

- **Project Manager**: Initiates the permit application and receives updates on permit status.
- **Government Authorities**: Review and decide on submitted permit applications.

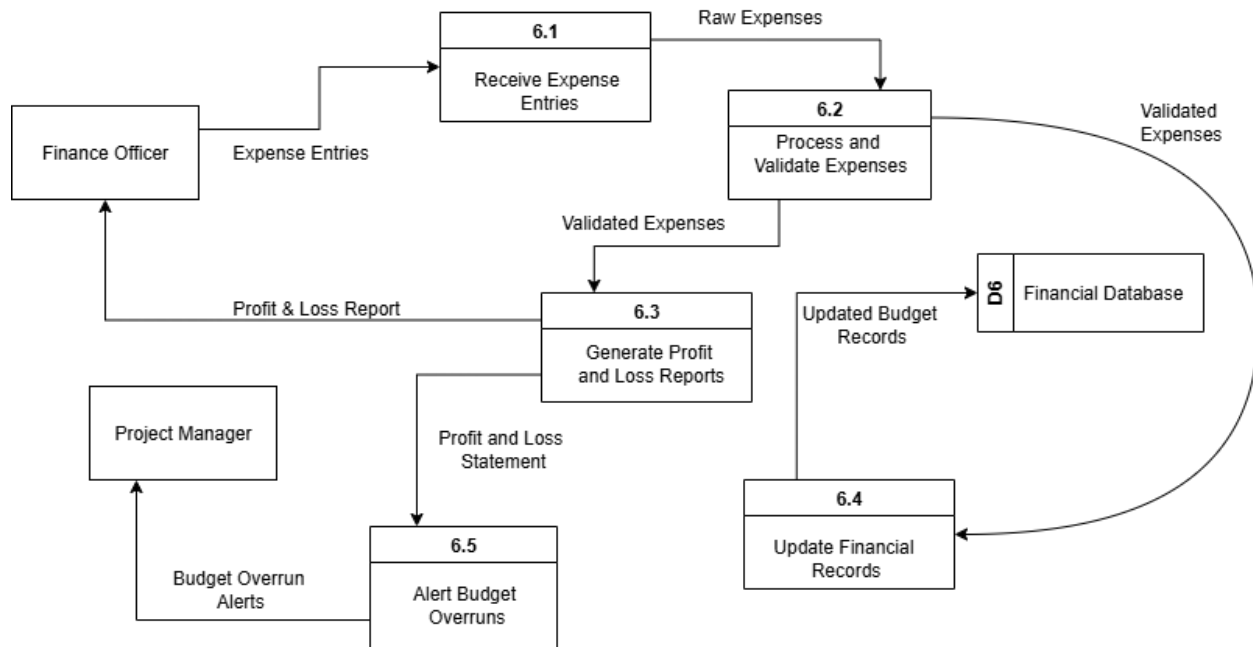
Data Store

- **D5 – Permit Database**: A centralized storage system that holds all submitted and finalized permit application records.

Data Flows

- **New Permit Application** is prepared and submitted to authorities.
- **Approval/Rejection Decision** is processed and used to update the system.
- **Permit Records** is stored in the **Permit Database**, and alerts are sent to stakeholders.

Child 6 diagram



This diagram expands on the “**Manage Project Finances**” process from the parent (Level 0) DFD. It outlines the steps involved in handling project expenses—from entry and validation to reporting and budget overrun detection.

Processes Explained

Process 6.1 – Receive Expense Entries

- **Input:** **Expense Entries** from the **Finance Officer**.
- **Function:** Records raw expense data submitted for the project.
- **Output:** Forwards **Raw Expenses** to the next process for validation.

Process 6.2 – Process and Validate Expenses

- **Input:** Raw expense data from 6.1.

- **Function:** Reviews and validates each expense to ensure accuracy and legitimacy.
 - **Output:**
 - Stores **Validated Expenses** into **D6 – Financial Database**.
 - Passes **Validated Expenses** to reporting (6.3) and record updating (6.4).
-

Process 6.3 – Generate Profit and Loss Reports

- **Input:** Validated expenses.
 - **Function:** Compiles financial reports including project **profit and loss statements**.
 - **Output:**
 - Sends **Profit & Loss Report** to the **Finance Officer**
 - Forwards **Profit and Loss Statement** to 6.5 for budget analysis.
-

Process 6.4 – Update Financial Records

- **Input:** Validated expense data.
 - **Function:** Updates budget and financial records to reflect recent transactions.
 - **Data Store:** Saves the **Updated Budget Records** into **D6 – Financial Database**.
-

Process 6.5 – Alert Budget Overruns

- **Input:** Profit and Loss Statement from 6.3.
- **Function:** Analyzes the financial report to identify if expenses exceed budget.

- **Output:** Sends **Budget Overrun Alerts** to the **Project Manager** for corrective action.
-

External Entities

- **Finance Officer:** Submits expense entries and receives financial reports.
 - **Project Manager:** Receives **budget overrun alerts** and **profit/loss updates** to make informed decisions.
-

Data Store

- **D6 – Financial Database:** Central repository that stores:
 - Validated expenses
 - Budget records
 - Financial updates for audit and tracking
-

Data Flows

- **Expense entries** flow from the Finance Officer to 6.1.
- **Validated expenses** move into the database and other financial processes.
- **Reports and alerts** are delivered to decision-makers for monitoring and control.

8.0 System Wireframe

Before implementing the coding for the system, we had to design the system interface first. Hence, we had designed the prototype design for the project management system by using Figma. The design for the interface for input design and output design are carefully considered to ensure a comfortable and intuitive interface for the user.

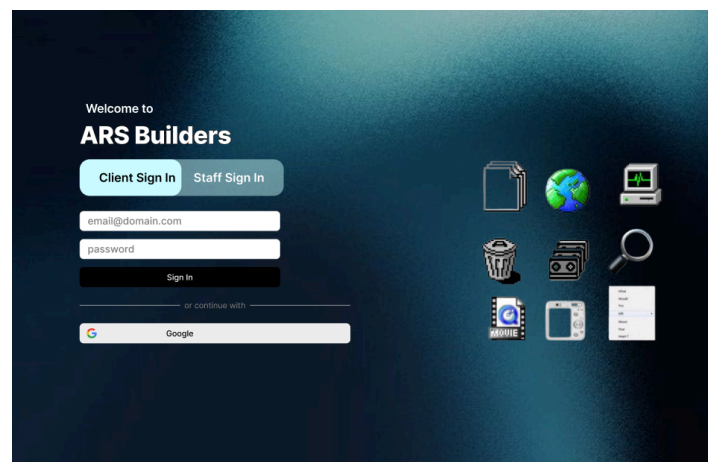
Input design :

1.User login Page

The Login page is used to enter information to log into the system. The components include:

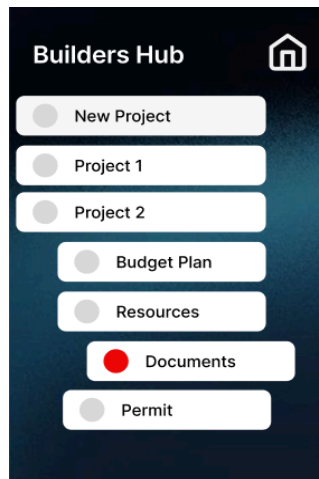
- Email / Username input
- Password input (with option to show/hide)
- Login button
- Password recovery link
- These input components are usable and facilitate secure access.

The Login Page is the point where user credentials are authenticated before proceeding into the ARS Builders Project Management System. The login page serves as the security bridge to make sure that only authorized users can access the system.



2.Navigation Sidebar

- It's located on the left side of the layout.
- Provides quick references to main sections such as Dashboard, Projects, Labour Management, Equipment Tracker, Material Inventory, Reports, and Settings.
- Navigation remains consistent throughout all the pages in the website.



3.Top Navigation Bar

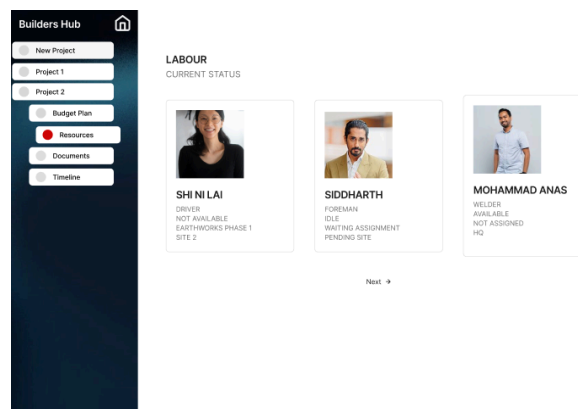
- Located at the top of the top of each screen for functionality specific to the user.
- Notifications, profile management, as well as a logout button features in the top Navigation Bar.



Output design :

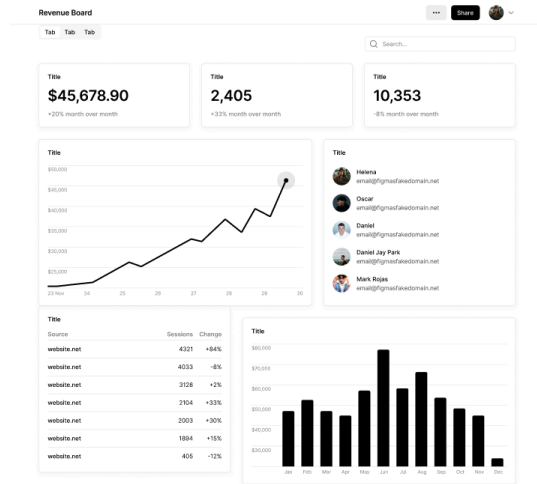
1.Main Content Area

- Provides dynamic type of content based on the module selected.
- For example, Labour Management will contain features such as charts and reporting, attendance, performance measures, and labour allocation chart.
- The main content area is modular and consists of cards, tables, and widgets, to allow Users to display what they would like, in a clear manner.



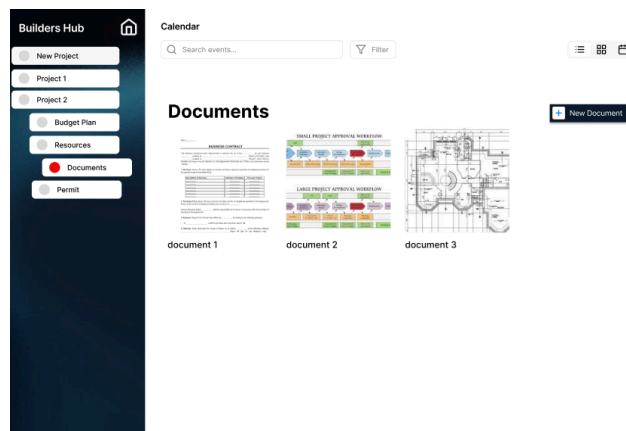
2.Dashboard Overview

- The landing page of the system gives an excellent overview of project health, resource usage, recent activity, and any overall metrics the users want to display.
- The dashboard consists of widgets that may graphically represent budget usage, task status comparison, or project progress.



3.Consistency and Usability

- Everything is set out in a grid, helping the users to easily understand while allowing for flexibility.
- Common UI patterns like buttons or dropdowns or filters will be used frequently on pages for consistency, to help improve usability, and so the user can quicker find the same patterns across the system.



Figma link :

<https://www.figma.com/design/RjWaR7BfC8j0EpYPoine6D/Untitled?node-id=0-1&t=rt24ZR3q6Lmkceg1-1>

9.0 Summary of the proposed system

To ensure the design of the new system fulfills the requirements of the client, we have analysed the requirement and designed a system that may help in reducing the project management loads and cost. The system is also designed with a user-friendly interface to ease the user in adapting to the new system and also to avoid the new system from interfering with the existing project management process.

After producing the wireframe of the new system, we may propose the design of the new system to the client to ensure the design is accepted by the client before implementation of the new system. After the design is approved, the implementation process may start and the production of the new system will begin.