

# Functional Design Specification (FDS)

## 1. Introduction

This Functional Design Specification (FDS) outlines how the system described in the User Requirement Specification (URS) for the DC Tracking Activity Tool will be functionally implemented. The tool is designed to support data centre project tracking, task management, time tracking, reporting, and financial oversight for Atlantach Technical Services.

## 2. System Overview

The system will be a cloud-based application accessible via web login. It supports multiple user roles, task tracking, time management, and progress reporting. The system integrates with existing pricing models and commissioning tools via APIs.

## 3. User Roles and Access Levels

- User: Submit field data
- Discipline Lead: Approve and review field activities
- Finance: View and generate financial reports
- Program Manager: Full access to all levels
- Senior Management: Full access
- Administrator: User management, configuration, and approvals

## 4. Functional Requirements Implementation

Each requirement from the URS is implemented as follows:

4.1 Access & Configuration: The system supports remote and local access with configurable profiles and user roles. Minimum two users per role to ensure redundancy.

4.2 Project Setup: Import project data (assets, tasks, time estimates). Weekly task assignment and time tracking features included.

4.3 Task Tracking: Field users can assign start and finish times. All actions are timestamped and audit-trailed.

4.4 Reporting: Automated CSV reports (daily/weekly/monthly) show completed, missed, and delayed tasks, time/cost deviations, and failed tests.

4.5 Revenue Recognition: Weekly task reports for WIP and revenue recognition. OOS verification workflow included.

4.6 Security: Microsoft 365-based security with cloud-based storage and access control.

4.7 API Integration: Interfaces with pricing and commissioning tools via APIs. Compatible with Atlantach IT cloud infrastructure.

4.8 Maintenance: Regular backups, 24/7/365 optional support, and vendor training.

4.9 Additional Features: Equipment tracking, customer access to asset status, alarm notifications, and record attachments per tag.

## **5. Validation Strategy**

The system will undergo Installation Qualification (IQ), Operational Qualification (OQ), and a Go Live signoff process. Performance Qualification (PQ) is not required as per the URS.

# Project Proposal: DC Tracking Activity Tool

## 1. Introduction

This proposal outlines the development and implementation of a DC Tracking Activity Tool for Atlantach Technical Services. The tool will enable effective project tracking, resource management, reporting, and invoicing for data centre operations.

## 2. Objectives

- Track project tasks and associated timelines
- Support In Scope (IS) and Out of Scope (OOS) tracking
- Enable real-time field updates
- Automate progress reporting
- Facilitate accurate revenue recognition and invoicing

## 3. Project Scope

The system will support field teams, program managers, finance, HR, and operations. It will include configurable roles, project setup tools, task assignment, time tracking, and API integration with existing systems.

## 4. Solution Overview

A cloud-based web application, secured with Microsoft 365, accessible on and off site. The system will include user role configuration, project import/export, audit trail logging, and customizable reporting tools.

## 5. Key Benefits

- Real-time visibility of project progress
- Improved accuracy in cost and time tracking
- Reduced manual effort in reporting
- Enhanced invoicing and revenue workflows
- Scalable to future data centre projects

## 6. Project Timeline (Estimate)

Phase 1: Requirements & Design - 2 weeks

Phase 2: Development - 4 to 6 weeks

Phase 3: Testing (IQ/OQ) - 2 weeks

Phase 4: Deployment & Go Live - 1 week

Total Estimated Duration: 9 to 11 weeks

## **7. Resources Required**

- 1 Project Manager
- 1 Full Stack Developer
- 1 QA Tester
- 1 Business Analyst
- Stakeholder availability for validation

## **8. Risk Management**

- Data sync issues: mitigated by automated backups and audit trail
- User adoption: mitigated by training and phased rollout
- Integration failures: addressed with early API testing

## **9. Estimated Cost (To Be Confirmed)**

Cost to be determined based on vendor quotes and internal resource allocation. A detailed cost analysis will follow during the requirement refinement phase.

## **10. Conclusion**

This project will enable Atlantach to manage its data centre activities more efficiently and position itself for scalable growth. Approval of this proposal will initiate the detailed planning and development phases.

## System Architecture Summary: DC Tracking Activity Tool

### 1. Overview

The DC Tracking Activity Tool is a cloud-based system designed for real-time project monitoring, reporting, and cost tracking. It is accessible through a secure web interface and supports multiple user roles. The system integrates with external pricing and commissioning tools via APIs, and it leverages Microsoft 365 authentication and cloud infrastructure for security and scalability.

### 2. Key Components

- Web App UI: Interface for users to log in, manage tasks, and view reports
- Application Logic Layer: Handles business rules, task processing, audit trails, and reporting logic
- Database: Stores project data, user roles, time logs, and change history
- API Layer: Enables integration with external systems like pricing models and commissioning tools
- Security Layer: Uses Microsoft 365 for authentication and access control
- Cloud Infrastructure: Hosted on Atlantachs IT cloud platform

### 3. Workflow Summary

Users access the system through a secure login. Field users submit task updates, which are processed by the application logic and stored in the database. Reports are generated automatically or on-demand. Integration points provide real-time synchronization with external pricing tools. Admins can configure user roles, projects, and access levels through the interface.

### 4. Architecture Diagram

Please refer to the attached system architecture diagram for a visual representation of component relationships.

