**dd**

**High Level Design Document**

Version: 1.0

Date: April 10, 2025

Author: Architecture Team

Table of Contents

Right-click to update table of contents.

# Introduction

## Purpose

Project Management feature with Tasks. Ability to have multiple customers. Customers would be all over the United States. They can schedule time also.

## Scope

I need an application for Residential Electricians that they can create projects and tasks for customers. The electrician would install outlets and perform other tasks. We want to use AWS, Lambda, S3, AIML.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| HLDD | High Level Design Document |
| API | Application Programming Interface |
| DB | Database |

# System Overview

Project Management feature with Tasks. Ability to have multiple customers. Customers would be all over the United States. They can schedule time also.

## System Context

This section describes the system context and its interactions with external systems and users.

## System Functions

* Project Management
* Scheduling
* Store details
* Menu
* Rating

## User Roles

|  |  |
| --- | --- |
| **Role** | **Description** |
| Administrator | Full system access |
| User | Standard application access |

# Architecture Overview

Scalable and maintainable architecture design

## Architecture Principles

* Scalability: Designed to handle growing user base
* Reliability: Robust error handling and redundancy
* Security: Comprehensive protection mechanisms

## Architecture Diagram

The following diagram illustrates the high-level architecture of the system:

[Architecture Diagram Placeholder]

## Component Descriptions

### Frontend

User interface layer

### Backend

Business logic and data processing

### Database

Data storage and retrieval

# Technology Stack

Modern, scalable technology stack

## Frontend

|  |  |  |
| --- | --- | --- |
| **Technology** | **Version** | **Purpose** |
| React | Latest | User interface framework |

## Backend

|  |  |  |
| --- | --- | --- |
| **Technology** | **Version** | **Purpose** |
| Python (Flask/Django) | Latest | Server-side logic and API |

## Database

|  |  |  |
| --- | --- | --- |
| **Technology** | **Version** | **Purpose** |
| PostgreSQL | Latest | Primary data storage |

# Data Architecture

This section describes the data architecture of the system including data models, storage, and flows.

## Data Stores

### Primary Database

**Type:** Relational **Purpose:** Main application data

### Cache

**Type:** In-memory **Purpose:** Temporary data caching

## Data Models

The following data models are used in the system:

[Data Model Diagram Placeholder]

## Data Flows

The following diagram illustrates the data flows in the system:

[Data Flow Diagram Placeholder]

# Security Architecture

Enhanced security for moderately sensitive applications

## Authentication and Authorization

Multi-factor authentication

## Data Protection

Advanced encryption (AES-256)

## Network Security

Network segmentation, advanced firewall rules

## Compliance

The system is designed to comply with relevant regulatory requirements.

# Deployment Architecture

Cloud-native deployment strategy

## Deployment Diagram

The following diagram illustrates the deployment architecture of the system:

[Deployment Diagram Placeholder]

## Environments

### Development

Local development environment

### Staging

Pre-production testing environment

### Production

Live production environment

## CI/CD Pipeline

The system uses a CI/CD pipeline for automated testing and deployment.

# Operational Considerations

This section describes the operational considerations for the system.

## Monitoring and Logging

The system implements monitoring and logging mechanisms to ensure operational visibility.

## Backup and Recovery

The system implements backup and recovery mechanisms to ensure data durability.

## Scaling and Performance

The system is designed to scale to handle increasing load and maintain performance.

## Disaster Recovery

The system implements disaster recovery measures to ensure business continuity.

# Appendices

## References

* Reference 1
* Reference 2
* Reference 3

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 2025-04-10 | Architecture Team | Initial version |