Lead Management Application Design Document

Introduction

This document provides a comprehensive design for the Lead Management Application, which allows users to create, get, and update leads. The application is built using FastAPI and includes features such as email notifications and authentication for accessing internal resources.

Functional Requirements

1. Create Lead: A publicly available form for prospects to fill in with fields: first name, last name, email, and resume/CV.
2. Get Leads: An internal endpoint to retrieve a list of all leads, accessible only to authenticated users.
3. Update Lead State: An internal endpoint to update the state of a lead, transitioning from PENDING to REACHED_OUT, accessible only to authenticated users.
4. Email Notifications: Upon creating a lead, emails are sent to both the prospect and an attorney within the company.

Technical Requirements

1. Framework: FastAPI
2. Database: SQLite for persistence
3. Authentication: Basic authentication for internal endpoints
4. Email Sending: Integration with an SMTP server to send emails

System Design

Architecture Overview

The application is designed as a RESTful API using FastAPI. It includes the following key components:

1. API Endpoints:
    - POST /leads/: Create a new lead
    - GET /leads/: Retrieve all leads (requires authentication)
    - PUT /leads/{lead_id}: Update the state of a lead (requires authentication)
2. Database:
    - SQLite database to store lead information
    - SQLAlchemy ORM for database interactions
3. Email Sending:

    - Use the aiosmtplib library to send emails asynchronously

4. Authentication:
    - Basic authentication for internal endpoints

Database Schema

The database schema consists of a single table leads with the following columns:

- id: Integer, Primary Key
- first_name: String
- last_name: String
- email: String
- resume: String
- state: Enum (PENDING, REACHED_OUT)