Software Requirements Specification (SRS)

Project Title: Mass Mailing Lead Generation Solution

Sponsored by: SourceRight Technologies(India) Pvt. Ltd.

1. Introduction

1.1 Goal

This document's goal is to outline the Mass Mailing Lead Generation Solution's functional and non-functional requirements.
This project aims to design and develop a cloud-based platform that allows Small and Medium Enterprises (SMEs) to automate email-based lead generation with high deliverability, AI-powered personalization, and cost-efficient scalability.

To guarantee a clear understanding between the development team and the project sponsor, the SRS offers a thorough explanation of the system's features, architecture, technology stack, and intended use cases.

1.2 Scope

Due to high expenses, a lack of technical resources, and complicated deliverability management, SMEs frequently find it difficult to effectively manage email campaigns.
In order to solve those issues, this solution offers a scalable, self-managed, AI-enhanced mass mailing system that combines lead intelligence, deliverability protection, and campaign management onto a single platform.

The scope consists of:

- Creating, planning, and sending campaigns with several SMTP providers.
- Management of domain reputation and authentication (SPF, DKIM, DMARC).
- Send-time optimization, lead scoring, and email personalization powered by AI.
- Monitoring of open, click, bounce, and reply metrics in real time.
- Basic CRM integration and dashboard visualization.

The system will be hosted on the AWS cloud infrastructure and be reachable through a web application.

1.3 Definitions, Acronyms, and Abbreviations.

Term	Definition
SMTP	Simple Mail Transfer Protocol, used for sending emails.
SPF/DKIM/DMARC	Email authentication standards for verifying sender identity.
AWS SES	Amazon Simple Email Service, cloud-based mail delivery tool.
AI	Artificial Intelligence, used for content personalization and optimization.
CRM	Customer Relationship Management system.
SME	Small and Medium Enterprise.

1.4 References

- Instantly.ai Platform (for comparative study).
- AWS SES and Route53 Documentation.
- FastAPI and React.js official documentation.
- MongoDB Atlas and Redis documentation.

1.5 Overview

This SRS document includes:

- A system overview
- Functional and non-functional requirements
- System architecture
- External interfaces
- Constraints and dependencies
- Expected outcomes

2. Overall Description

2.1 Product Perspective

This system is an independent, modular, cloud-hosted application.
It acts as a bridge between basic email marketing tools and enterprise-grade marketing platforms, enabling SMEs to perform scalable lead generation while maintaining sender reputation.

The architecture follows a client–server model, with:

- Frontend: React.js + Bootstrap (for UI)
- Backend: FastAPI (Python)
- Database: MongoDB Atlas
- Mail Infrastructure: AWS SES and SendGrid
- Cloud Services: AWS EC2, S3, Route53
- AI Engine: For personalization, lead scoring, and timing optimization

2.2 Product Functions

Key functions of the system include:

1. **User Management:**
   User management includes role-based access control, authentication (JWT), and registration.

2. **Campaign Management:**
   - Plan and organize large-scale email campaigns.
   - Provide recipient lists and templates.
   - Automate campaign performance monitoring.

3. **Deliverability Management:**
   - Handle several domains with deliverability management.
   - Set up DMARC, DKIM, and SPF records.
   - Track domain reputation and blacklist status in real time.

4. **AI-Powered Optimization:**
    - Customize content blocks and subject lines.
    - Predict optimal send times based on engagement data.
    - Prioritize potential clients by assigning lead scores.

5. **Infrastructure Scalability:**
    - Support multi-SMTP sending.
    - Use AWS to automatically scale your computing resources.

2.4 Constraints

- Requires stable internet connectivity for cloud services.
- Dependent on SMTP provider APIs (AWS SES, SendGrid).
- AI features depend on historical data volume for learning patterns.
- Email sending is restricted by provider quotas and domain reputation.

2.5 Assumptions and Dependencies

- Users have verified sender domains.
- SMTP credentials and API keys are valid.
- AWS and MongoDB Atlas services remain available.
- Project phases will follow Agile iterations.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication

- The system shall allow users to register and log in using JWT-based authentication.
- Passwords shall be securely hashed using bcrypt.

3.1.2 Campaign Management

- The system shall allow users to create campaigns with name, subject, body, recipient list, and schedule.
- Users shall be able to edit or delete existing campaigns.
- Campaigns shall be sent via connected SMTP providers (AWS SES, SendGrid).

3.1.3 Deliverability Management

- The system shall manage SPF, DKIM, and DMARC configurations.
- It shall monitor blacklist status using DNSBL APIs.
- It shall send alerts if a domain's reputation drops below threshold.

3.1.4 AI Optimization

- The system shall use AI to suggest optimal send times based on previous engagement data.
- It shall recommend personalized content blocks for different lead categories.
- It shall assign scores to leads based on interaction metrics.

3.1.5 Analytics & Reporting

- The system shall track open, click, bounce, and reply events via SMTP webhooks.
- The dashboard shall display summarized analytics in graphical format.

3.1.6 Data Management

- The system shall store all campaign, lead, and analytic data in MongoDB.
- It shall provide API endpoints for data retrieval and dashboard visualization.

3.2 Non-Functional Requirements

Category	Requirement
Performance	The system should handle at least _____ emails per campaign with queue-based sending.
Scalability	Auto-scaling architecture via AWS EC2.
Security	All APIs secured with HTTPS; JWT for sessions; bcrypt for password hashing.
Reliability	99% uptime target using AWS cloud services.
Usability	Clean, responsive UI using Bootstrap.
Compliance	GDPR and CAN-SPAM regulations are enforced.
Maintainability	Modular code structure with separate backend and frontend repos.
Portability	Docker containers ensure cross-environment compatibility.

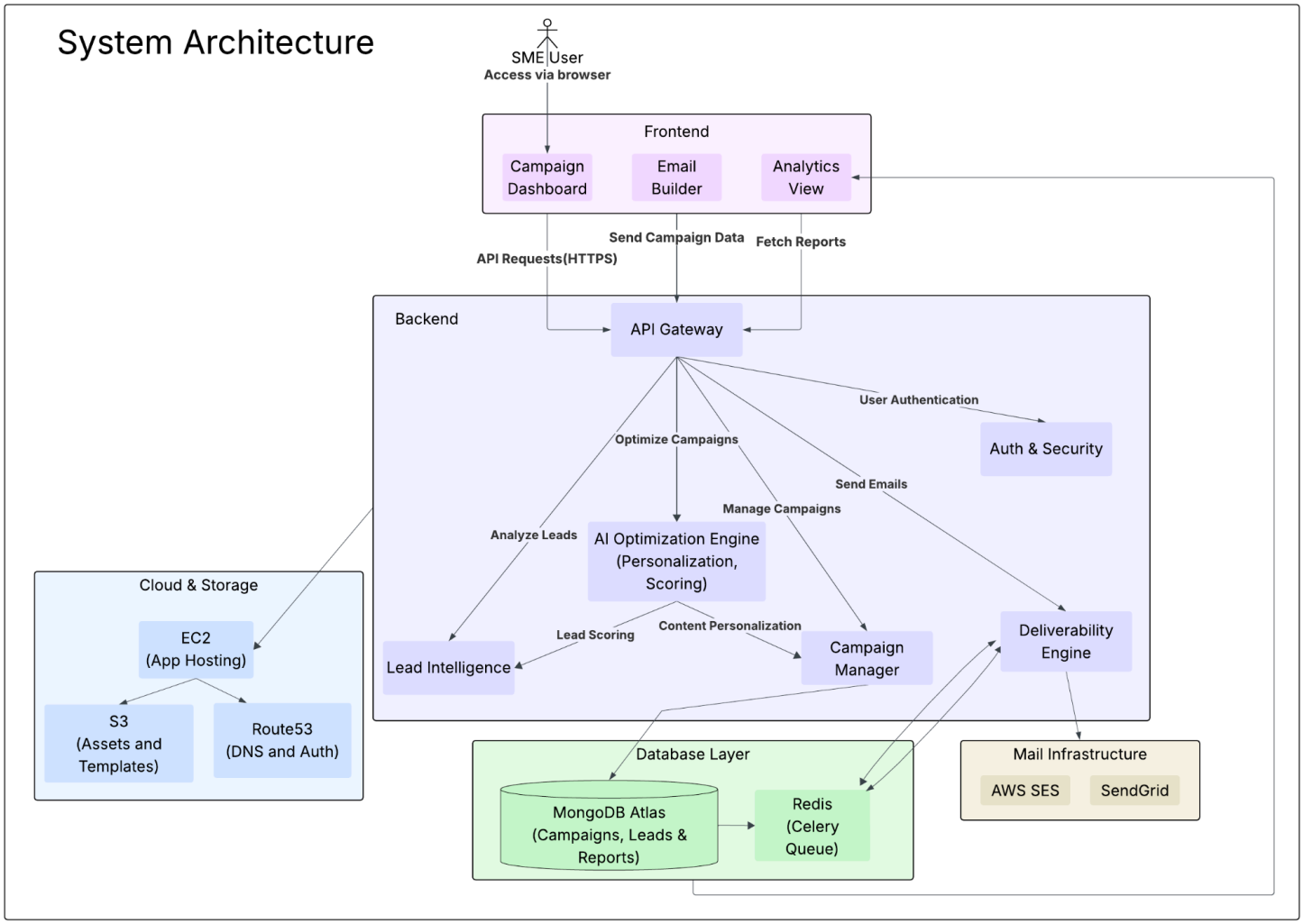
3.3 External Interface Requirements

Interface Type	Description
User Interface	Web-based React app accessible via browsers.
API Interface	RESTful endpoints exposed by FastAPI backend.
Database Interface	MongoDB Atlas using PyMongo driver.
SMTP Interface	AWS SES and SendGrid SMTP or REST APIs.
Cloud Interface	AWS EC2, S3, and Route53 integrations.

4. System Architecture

4.1 Overview

The system follows a modular microservice-inspired architecture, separating the frontend, backend, and infrastructure for scalability and maintainability.



Layers include:

1. **Frontend:** React + Bootstrap
2. **Backend:** FastAPI (Python)
3. **AI Engine:** Personalization, send-time optimization, lead scoring
4. **Database:** MongoDB Atlas
5. **SMTP Providers:** AWS SES and SendGrid
6. **Cloud Services:** AWS EC2 (hosting), S3 (assets), Route53 (DNS)

5. Expected Outcomes

- A functional web-based demo that can create, schedule, and send email campaigns.
- Real-time campaign analytics with deliverability and reputation metrics.
- AI-assisted campaign optimization (timing, personalization, and scoring).
- Cost-efficient, cloud-deployable infrastructure ready for SME adoption.

6. Future Enhancements

- Integration with additional SMTP providers (Mailgun, Postmark).
- Advanced AI-based sentiment analysis on replies.
- Mobile application for campaign tracking.
- Voice-based AI assistant for lead insights.

7. Conclusion

By bridging the gap between enterprise SaaS platforms and simple email tools, the Mass Mailing Lead Generation Solution seeks to offer SMEs a robust yet reasonably priced platform.

Cloud-based scalability, AI-driven intelligence, and strong deliverability features enable smaller businesses to compete successfully in digital marketing while keeping complete control over their expenses and infrastructure.