Product Requirements Document (PRD) for Frontend Using Spring Thymeleaf

1. Project Overview

- The frontend will utilize Spring Thymeleaf as a server-side template engine to render dynamic web pages for the Rawasy WhatsApp CRM.
- It will provide a seamless user experience for customer relationship management, marketing automation, and customer support.

2. Architecture & Technology

- **Framework**: Spring Boot with Thymeleaf
- **Frontend**: HTML/CSS with Thymeleaf templates
- **Backend**: Java 17, Spring Boot, JPA/Hibernate, MySQL
- **API**: RESTful services for data interaction

3. Key Features

- Dynamic rendering of web pages using Thymeleaf templates.
- Integration with backend RESTful APIs for data fetching and submission.
- Role-based access control for different user roles (Admin, Marketing, Sales, Support).
- User-friendly forms for contact uploads, campaign management, and ticket handling.
- Responsive design for compatibility across devices.

4. Functional Requirements

- Render user interfaces for managing users, roles, and campaigns.
- Display dynamic content based on user roles and permissions.
- Implement forms for data input with validation.
- Provide feedback and notifications to users based on actions taken.

5. Non-Functional Requirements

- Ensure fast loading times and responsive design.
- Maintain security best practices for user data handling.
- Optimize for SEO with proper HTML structure.

Product Requirements Document (PRD) for Frontend Using Angular

1. Project Overview

- The frontend will leverage Angular to create a single-page application (SPA) for the Rawasy WhatsApp CRM.
- It will enhance user interaction and provide a modern web experience for managing customer relationships.

2. Architecture & Technology

- **Framework**: Angular
- **Frontend**: TypeScript, HTML, CSS
- **Backend**: Java 17, Spring Boot, JPA/Hibernate, MySQL
- **API**: RESTful services for data interaction

3. Key Features

- SPA architecture for a smooth user experience without full page reloads.
- Integration with backend RESTful APIs for real-time data updates.
- Role-based access control for different user roles (Admin, Marketing, Sales, Support).
- Dynamic forms for contact uploads, campaign management, and ticket handling.
- Responsive design for compatibility across devices.

4. Functional Requirements

- Implement components for managing users, roles, and campaigns.
- Use Angular services to interact with backend APIs.
- Provide user feedback through notifications and alerts.
- Implement routing for navigation between different sections of the application.

5. Non-Functional Requirements

- Ensure fast performance and responsiveness.
- Follow security best practices for API interactions.
- Optimize for SEO with Angular Universal if server-side rendering is required.

Product Requirements Document (PRD) for Coupling Thymeleaf and Angular

1. Project Overview

- This document outlines the integration of Spring Thymeleaf and Angular to create a hybrid frontend for the Rawasy WhatsApp CRM.
- The goal is to leverage the strengths of both technologies while maintaining the same business logic across the application.

2. Architecture & Technology

- **Frameworks**: Spring Boot with Thymeleaf and Angular
- **Frontend**: HTML/CSS with Thymeleaf templates and Angular components
- **Backend**: Java 17, Spring Boot, JPA/Hibernate, MySQL
- **API**: RESTful services for data interaction

3. Key Features

- Use Thymeleaf for server-side rendering of initial pages and Angular for dynamic content updates.
- Maintain a consistent business logic layer across both frontend technologies.
- Role-based access control for different user roles (Admin, Marketing, Sales, Support).
- Dynamic forms for contact uploads, campaign management, and ticket handling.

4. Functional Requirements

- Render initial pages with Thymeleaf and load Angular components for dynamic interactions.
- Implement shared services for data fetching and business logic.
- Provide user feedback through notifications and alerts in both frameworks.
- Ensure seamless navigation between Thymeleaf-rendered pages and Angular components.

5. Non-Functional Requirements

- Ensure fast performance and responsiveness across both frameworks.
- Follow security best practices for user data handling and API interactions.
- Optimize for SEO with proper HTML structure and Angular Universal for server-side rendering.