

# Software Design Document (SDD) for LinkedIn Platform

Heramb Patil

March 2025

## 1 Introduction

### 1.1 Design Overview

The LinkedIn-like platform enables users to create professional profiles, share posts, send messages, and apply for jobs. The design follows a monolithic architecture using Laravel for both the backend and frontend:

- **Presentation Layer:** Laravel Blade templates (Frontend)
- **Application Layer:** Laravel (Backend APIs and Logic)
- **Data Layer:** MySQL Database

Key Features:

- User authentication and profile management
- Posting and sharing professional updates
- Premium user features: InMail, job insights
- Recruiter functionalities: job postings
- Admin management for reviewing users

### 1.2 Requirements Traceability Matrix

Requirement ID	Requirement Description	Design Component
R1	User Login and Registration	Authentication Service
R2	Create and Share Posts	Post Management Module
R3	Send InMail (Premium)	Messaging Service
R4	View Job Insights (Premium)	Job Insights Module
R5	Post Job Listings (Recruiter)	Job Management Module
R6	Admin: Review and Block Users	Admin Panel

## 2 System Architecture Design

### 2.1 Chosen System Architecture

The system follows a monolithic architecture with the following components:

- **Frontend and Backend:** Laravel Blade templates (Frontend) with integrated RESTful APIs (Backend)
- **Database:** MySQL for relational data storage, Redis for caching
- **External Interfaces:** Payment gateway API (Stripe), Email/SMS notifications (Nodemailer)

### 2.2 System Interface Description

Interface	Description
Operating System	Ubuntu-based cloud deployment
Database Interface	MySQL via Eloquent ORM
Network Interface	HTTPS/REST over TCP/IP
UI Framework	Laravel Blade Templates, Bootstrap
External Libraries	Stripe SDK, JWT, Nodemailer

## 3 Detailed Component Description

### 3.1 Authentication Service

- Responsibilities: User registration, login, JWT token authentication
- Constraints: Secure password storage (bcrypt), session expiration
- Resources: User database, Redis session cache

### 3.2 Post Management Module

- Responsibilities: Create, edit, delete posts
- Constraints: Must handle high traffic efficiently
- Resources: Post database with indexing for fast retrieval

### 3.3 Messaging Service (Premium Feature)

- Responsibilities: Handle InMail messaging
- Constraints: Only accessible by premium users
- Resources: Message database, WebSocket for real-time updates

## 4 User Interface Design

### 4.1 Description of the UI

The UI follows a modern responsive design with clear navigation:

- **Login Page:** Email, password fields, login button
- **Home Feed:** Displays posts with like/comment functionality
- **Messaging:** Inbox, compose, and send messages
- **Job Section:** Browse and apply for jobs
- **Admin Panel:** View reported users and take action

### 4.2 User Actions and Events

Screen	Action	System Response
Login	Enter Credentials	Validate and redirect to homepage
Home Feed	Create Post	Save post and display on feed
Messaging	Send InMail	Store message and notify recipient
Job Section	Apply for Job	Save application in database
Admin Panel	Block User	Update user status in DB

## 5 System Architecture Overview

### 5.1 System Partitioning

- **UI Layer:** Handles user interactions via Laravel Blade templates
- **Business Logic Layer:** Processes requests, validates data within Laravel controllers
- **Data Layer:** Stores user and platform data within MySQL, caching with Redis
- **External Services:** Payment processing, notifications via external APIs

### 5.2 Use Case: Post Creation

- **Actors:** Registered User
- **Preconditions:** User is logged in
- **Trigger:** User submits a new post
- **Normal Flow:** Post → Controller → Store in DB → Confirm to user
- **Alternative Flows:** Server error → Retry or show failure message
- **Business Rules:** No inappropriate content

## 6 Data Flow Specifications

### 6.1 Level 0 DFD: Context Diagram

Entities interacting with the system:

- User – logs in, posts, messages
- Premium User – accesses InMail, job insights
- Recruiter – posts jobs
- Admin – moderates users

Processes:

- User Authentication
- Post Management
- Messaging System
- Job Listings

Data Stores:

- User DB
- Post DB
- Message DB
- Job Listings

### 6.2 Level 1 DFD: Messaging System

Sub-processes involved:

- Validate User Authentication
- Send/Receive Messages
- Store Messages in Database
- Notify Recipient