**HIGH LEVEL DESIGN DOCUMENT**

**Professional Social Networking Application**

**Document Version:** 1.0.0

**Date:** 03/03/2024

**Prepared by** Shashidhar P H

**Table Of Contents**

**1. Introduction**

- Purpose of the Document

- Scope of the Document

- Overview of the Social Media Platform

- Key Objectives and Goals of the System

**2. Requirements and Goals**

- Functional Requirements

- User Authentication and Authorization

- User Profile Management

- Content Creation and Publishing

- Social Interactions

- Messaging and Notifications

- Search and Discovery

- Non-Functional Requirements

- Performance

- Scalability

- Security

- Availability

- Reliability

- Usability

**3. Assumptions and Prerequisites**

- Technology Stack

- Infrastructure Requirements

- Data Privacy and Compliance

**4. Business Overview**

- Business Goals and Objectives

- Target Audience

- Competitive Analysis

**5. High-Level Design**

- Architectural Overview

- System Components

- Communication Protocols

- Deployment Architecture

- Data Model

- Entity Relationship Diagram (ERD)

- Data Storage Strategy (Database Schema)

- Application Design

- User Interface Design

- Client-Server Interaction

- Server-Side Logic

- Caching Strategy

- Integration Points

- Third-Party Integrations

- API Design

- Security Design

- Authentication and Authorization Mechanisms

- Data Encryption

- Threat Modelling

- Performance Considerations

- Load Balancing

- Caching Strategies

- Content Delivery Network (CDN) Integration

**6. Application Modules**

- User Management Module

- Content Management Module

- Social Interaction Module

- Messaging Module

- Search and Discovery Module

**7. Transactions and User Flows**

- User Registration and Authentication Flow

- Profile Setup Flow

- Content Creation Flow

- Social Interaction Flow

- Messaging Flow

- Search and Discovery Flow

**8. Future Enhancements**

- Feature Roadmap

- Technology Upgrades

- Scalability Improvements

**9. Conclusion**

- Summary of Key Points

- Next Steps

1. **Introduction**

**Purpose of the Document**

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

**Scope of the Document**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

**Overview of the Social Media Platform**

Social media is a collective term for websites and applications that focus on communication, community-based input, interaction, content-sharing, and collaboration. People use social media to stay in touch and interact with friends, family, and various communities. Businesses use social applications to market and promote their products and track customer concerns. In business, social media is used to market products, promote brands, connect to customers, and foster new business. As a communication platform, social media promotes customer feedback and makes it easy for customers to share their experiences with a company. Businesses can respond quickly to positive and negative feedback, address customer problems and maintain or rebuild customer confidence. Social media is also used for crowdsourcing. That's the practice of using social networking to gather knowledge, goods, or services. Companies use crowdsourcing to get ideas from employees, customers, and the public for improving products or developing future products or services.

**Key Objectives and Goals of the System**

This application will allow users to share photos and videos with other users. Additionally, users can follow other users based on follow request and they can see other user’s photos and videos. In this system, you can search users and see their profile if their account is public. Otherwise, you need to send a follow request. Users can be able to manage their profile they can edit their details likes address, interest, skills, etc at any time. And they even can see who liked their post and can see comments, user can even report abusive post and comments in application. Social networking application allow users to share ideas, activities, events, and interests within their individual networks.

**2. Requirements and Goals**

**Functional Requirements**

This is what the system must deliver. It may be said that this is the system’s primary objective.

* User Authentication and Authorization.
* User Profile Management
* Content Creation and Publishing
* Social Interactions
* Messaging and Notifications
* Search and Discovery
* Non-Functional Requirements
* Performance
* Scalability
* Security
* Availability
* Reliability
* Usability

**3. Assumptions and Prerequisites**

**Technology Stack:**

1. **Frontend Development:**
   * Framework: Angular or React for building responsive, interactive user interfaces.
   * Languages: HTML, CSS, JavaScript/TypeScript.
   * UI Libraries: Bootstrap or Material UI for styling and layout.
2. **Backend Development:**
   * Framework: Spring Boot for building RESTful APIs and microservices.
   * Language: Java for backend logic implementation.
   * Database: MySQL or MongoDB for storing user data.
   * Authentication: Spring Security for handling user authentication and authorization.

**Infrastructure Requirements:**

* + Mobile
  + PC or desktop
  + Good internet bandwidth
  + Tablet

**Data Privacy and Compliance**

* + Data Encryption
  + User Consent
  + Data Minimization
  + Secure Authentication
  + Third-Party Data Handling
  + Data Breach Response Plan

**4. Business Overview**

**Business Goals and Objectives:**

* + User Engagement
  + User Satisfaction
  + Security and Privacy
  + Enhance user interface, introduce interactive features, and optimize push notifications to keep users actively participating.
  + Implement targeted marketing strategies, referral programs, and partnerships to attract new users.
  + Introduce premium features, explore advertising opportunities, and implement a subscription model for enhanced services.
  + Increase the visibility and recognition of the social networking app.
  + Launch marketing campaigns, collaborate with influencers, and actively engage in public relations efforts.
  + Ensure high levels of user satisfaction with the social networking app.
  + Gather user feedback, address issues promptly, and continuously improve the app based on user preferences.
  + Prioritize the security and privacy of users on the social networking app.
  + Implement robust security measures, comply with data protection regulations, and educate users about privacy controls.
  + Facilitate user interactions, create forums or groups, and organize events or challenges to strengthen the user community.
  + Implement robust analytics tools, analyse user behaviour, and derive insights to enhance app performance and user experience.

**Target Audience:**

**1.** **Young Adults and Millennials:**

Age group: 18-34, Tech-savvy, socially active, and open to new trend.

**2. Professionals and Networking:**

Working professionals: Seeking career opportunities, professional networking, and industry insights.

1. **Students and Academia:**

Educational Level: High school, college, or university students can discuss about their academics, group projects, and socializing with peers.

**4. Tech Enthusiasts:**

Who is very fond of new technology, gadgets, and innovation. Can connect and discuss tech trends, share insights, and connect with professionals in the tech industry.

**Competitive Analysis:**

A competitive analysis for a social networking app involves evaluating key competitors in the market.

**LinkedIn:**

Strengths: Professional networking, job opportunities, business-focused content.

Weaknesses: Limited personal content sharing, less casual interaction.

**5. High-Level Design**

**Architectural Overview:**

**System Components:**

**Load balancer:**

A load balancer is a system design component that is used to distribute incoming requests or workloads across several different resources or servers.

**Deployment Architecture:**

**A diagram of a company

Description automatically generated**

**Software architecture:**

* + **Frontend Development:**
  + Framework: Angular or React for building responsive, interactive user interfaces.
  + Languages: HTML, CSS, JavaScript/TypeScript.
  + UI Libraries: Bootstrap or Material UI for styling and layout.
  + **Backend Development:**
  + Framework: Spring Boot for building RESTful APIs and microservices.
  + Language: Java for backend logic implementation.
  + Database: MySQL or MongoDB for storing user data.

**Data Storage Strategy (Database Schema):**

MySQLDatabases store user data such as user profiles, posts, comments, and connections. system architecture incorporates both relational databases and we will incorporate Amazon S3 (Simple Storage Service) to overcome size limitations and ensure high availability.

**Application Design:**

* + User Interface Design
  + Client-Server Interaction
  + Server-Side Logic
  + Caching Strategy
  + Integration Points
  + Third-Party Integrations
  + API Design
  + Security Design
  + Authentication and Authorization Mechanisms
  + Data Encryption
  + Threat Modelling
  + Performance Considerations
  + Load Balancing
  + Caching Strategies
  + Content Delivery Network (CDN) Integration

**6. Application Modules**

**User Management Module:** consists of

* + Login.
  + Registration
  + Authentication.
  + View profile
  + Save of post

**Content Management Module**:

* + Posts
  + Photos
  + Videos

**Social Interaction Module:**

* Likes
* Comments
* Private chats

**Messaging Module:**

* Group message
* Private message

**Search and Discovery Module:**

* Search for people
* Search for content
* Browse and filter.

**7. Transactions and User Flows:**

**A white paper with black writing on it

Description automatically generated**

**8. Future Enhancements**

**Feature Roadmap:**

**User Experience (UX) Enhancements:**

Regularly update and refine the app's user interface to improve overall user experience.Implement user feedback and conduct usability studies to understand and meet user expectations.

**Personalization and AI Integration:**

Explore AI-driven features for personalized content recommendations, friend suggestions, or targeted advertisements.Implement chatbots or virtual assistants to enhance user engagement and support.

**Privacy and Security Measures:**

Stay vigilant about user data privacy and security.Implement advanced encryption methods, secure authentication, and regularly audit security protocols.

**Integration with Emerging Technologies:**

Explore integration with emerging technologies such as augmented reality (AR) or virtual reality (VR) for unique and immersive experiences.Consider compatibility with wearable devices or smart home technologies.

**Content Diversification:**

Allow users to share a variety of content types, including multimedia, short-form videos, and interactive content. Integrate live streaming features to encourage real-time interaction.

**Community Building and Engagement:**

Foster community building through features like groups, forums, or events. Enhance engagement through gamification, challenges, or rewards programs.

**Cross-Platform Compatibility:**

Ensure the app is compatible with various platforms and devices to reach a broader user base.Optimize the user experience for both mobile and desktop environments.

**Scalability Improvements**

* **Distributed Architecture:**

Implement a microservices architecture to break down the application into smaller, independent services. Distribute services across multiple servers or even data canters to handle increased load.

* **Load Balancing:**

Use load balancers to evenly distribute incoming traffic across multiple servers. Ensure that no single server becomes a bottleneck by efficiently distributing the workload.

* **Caching Mechanisms:**

Implement caching strategies for frequently accessed data to reduce database load. Use in-memory caching solutions like Redis or Memcached to store and retrieve frequently accessed data quickly.

* **Database Optimization:**

Optimize database queries for efficiency. Consider database sharing or partitioning to distribute data across multiple database instances.

* **Asynchronous Processing:**

Offload resource-intensive tasks to background jobs using message queues. Implement asynchronous processing for tasks like image processing, notifications, or data analysis.

* **Content Delivery Network (CDN):**

Use a CDN to distribute static assets (images, videos) to servers geographically closer to users, reducing latency. Offload the serving of static content from your main servers to CDN servers.

* **Horizontal Scaling:**

Scale your application horizontally by adding more servers to the infrastructure. Use auto-scaling mechanisms to dynamically adjust resources based on demand.

* **Stateless Services:**

Design services to be stateless, allowing them to handle requests independently. Store session data in a centralized and scalable system, such as Redis**.**

* **Efficient Data Storage:**

Choose scalable and distributed storage solutions like Amazon S3 or Google Cloud Storage for large media files. Optimize data models to reduce redundancy and improve retrieval efficiency.

* **Monitoring and Scaling Policies:**

Implement robust monitoring tools to track system performance and identify bottlenecks. Set up scaling policies to automatically adjust resources based on predefined conditions (e.g., increased traffic).

* **Global Content Distribution:**

Utilize edge computing and replicate data closer to users in different geographical regions. Ensure that your app can seamlessly serve users from various locations.

* **Resilience and Failover Mechanisms:**

Implement redundancy and failover mechanisms to ensure continuous service even if a server or component fails. Design with fault tolerance in mind to prevent a single point of failure.

**9. Conclusion**

The success of our social networking app hinges on a strategic blend of user-centric design, cutting-edge technology integration, and a commitment to scalability and security. By prioritizing a seamless user experience, embracing emerging technologies, implementing robust scalability measures, and ensuring privacy and ethical considerations, we aim to create a vibrant and resilient platform. Regularly revisiting and adapting our strategies based on user feedback and industry trends will be pivotal in maintaining the app's relevance and impact in the dynamic landscape of social networking. Together, let's embark on this journey to build a socially engaging, innovative, and user-friendly platform that stands the test of time.

**Next Steps:**

**Our next step includes to lessen.**

* Spam and Unsolicited Messages
* invasion of privacy.
* identity theft.
* Building Bot free platform