



SOFTWARE DESIGN SPECIFICATION

React Js  
  
Linked-in Replica

|  |  |  |  |
| --- | --- | --- | --- |
| **Created By:** | Utekar Pranit Ganesh | **Approved By:** | Manikanth Babu |
| **Created On:** | 23-09-2023 | **Approved On:** | 25-09-2023 |

Page left blank intentionally

**INDEX**

[**1** **PURPOSE** 2](#_Toc142418236)

[**2** **PROJECT SCOPE** 2](#_Toc142418237)

[**3** **SYSTEM OVERVIEW** 2](#_Toc142418238)

[**4** **DESIGN CONSIDERATIONS** 2](#_Toc142418239)

[4.1 Requirements 3](#_Toc142418240)

[4.2 Assumptions 3](#_Toc142418241)

[4.3 Dependencies 3](#_Toc142418242)

[**5** **SYSTEM ARCHITECTURE** 3](#_Toc142418243)

[5.1 Architectural Strategies 4](#_Toc142418244)

[5.2 Structure & Relationships 4](#_Toc142418245)

[**6** **DETAILED DESCRIPTION OF COMPONENTS** 4](#_Toc142418246)

[**7** **INTEGRATION** 5](#_Toc142418247)

[**8** **APPENDICES** 1](#_Toc142418248)

[8.1 Appendix A – Detailed Description of Components 1](#_Toc142418249)

**General Instructions for using the Live Project POC Document**

* This template and the subsequent document created using this template is a confidential document and is the intellectual property of Cloud Counselage Pvt. Ltd. Circulating it outside of the organisation without the consent of Cloud Counselage Pvt. Ltd. is the breach of company policies and will lead to legal actions
* The Design Specification of a software forms the basis of development of software
* The **text between inequality (< >) is to be replaced** by relevant text
* Please **remove the yellow highlight on the Text** between the inequality (< >). This is done to help you notice the text to be changed/replaced
* The text in *italics* highlighted in grey is just for reference and should be removed after adding the relevant text

# **PURPOSE**

the purpose of the LinkedIn Replica project is to offer a professional networking platform that combines the familiarity of LinkedIn with innovative features, greater user control, and an enhanced commitment to privacy. This purpose aims to empower professionals to connect, collaborate, and thrive in their respective industries while maintaining control over their personal and professional information.

# **PROJECT SCOPE**

The scope of the "LinkedIn Clone" includes its distinct features, its benefits, and its limitations. The system's distinct features allow it to replicate the core functionalities of LinkedIn by using modern development tools and inputs. The system enables the user to create a professional networking platform that addresses the need for a privacy-focused, user-centric, and customizable alternative to LinkedIn.

Distinct Features:

Privacy Emphasis: The LinkedIn Clone places a strong emphasis on user privacy, allowing users to have greater control over their personal data and interactions.

User-Centric Design: The platform is designed with a user-centric approach, offering an intuitive and customizable user interface that caters to individual professional preferences.

Replication of Core Features: It replicates core features of LinkedIn, including user profiles, connections, job postings, messaging, events, and news feeds, providing a familiar user experience.

Innovative Additions: The platform introduces innovative features and functionalities that set it apart from LinkedIn, offering users unique value and capabilities.

Community Engagement: It fosters community engagement by encouraging users to actively participate in discussions, share insights, and collaborate within a vibrant professional network.

Benefits:

Data Privacy: Users can enjoy a heightened level of data privacy and control over their personal information and interactions.

User Satisfaction: The user-centric design enhances user satisfaction and engagement, making it a preferred choice for professionals.

Networking Opportunities: The platform facilitates networking among professionals, opening up opportunities for career growth and advancement.

Innovation: The introduction of innovative features keeps the platform fresh and competitive in the professional networking space.

Community Building: Users can engage with a diverse and active community, promoting knowledge sharing and collaboration.

Limitations:

Feature Parity: While replicating core LinkedIn features, there may be limitations in achieving feature parity, especially in advanced functionalities.

User Migration: Encouraging users to migrate from the original LinkedIn platform may be challenging due to established user habits.

Resource Constraints: The project may face limitations in terms of resources, potentially impacting the scope and scale of certain features.

In summary, the scope of the LinkedIn Clone project encompasses the development of a professional networking platform with distinct features that prioritize privacy, user satisfaction, and community engagement. It aims to deliver a viable alternative to LinkedIn while acknowledging certain limitations inherent in the project's execution.

# **SYSTEM OVERVIEW**

This section will provide an outline of the various components and subsystems of the "LinkedIn Clone" project.

**LinkedIn Clone Components and Subsystems:**

**User Management System:**

**User Registration**: Allows users to create accounts with their personal and professional information.

**Login and Authentication**: Manages user authentication and access to the platform.

User Profile Management:

**Profile Creation**: Enables users to create and customize their professional profiles.

Profile Editing: Allows users to update and modify their profile information, including personal details, skills, and work history.

Connections and Networking:

**News Feed:** Displays updates, posts, and content shared by connections and relevant to the user's interests.

**Content Sharing:** Allows users to create and share professional content, such as articles, posts, and media.

Job Portal:

**Job Postings**: Lists job opportunities and allows users to search, apply, and receive job recommendations.

Job Application Management: Provides tools for managing job applications and tracking application status.

**Events**: Allows users to discover and participate in professional events, webinars, and conferences.

Networking Groups: Facilitates the creation and management of professional networking groups and communities.

**Privacy and Security:**

**Data Privacy**: Implements robust data privacy measures and user data protection.

Security Measures: Ensures platform security against cyber threats and unauthorized access.

# **DESIGN CONSIDERATIONS**

This section describes requirements, assumptions and dependencies to be addressed to devise a complete design solution.

## Requirements

* Functional Requirements

1. User Registration and Authentication 2) User Profiles 3) Job Postings and Application System 4) Security and Privacy

* **Non Functional Requirements**

1 ) Performance - Response Time, Scalability, Availability 2) Security- Data Encryption, Authentication and Authorization, Data Protection, User Privacy 3) Usability- User Interface (UI) Consistency, Mobile responsive, Accessibility

## Assumptions

1) Resource Availability: The availability of required resources, including human resources and technology infrastructure, is assumed for project execution. 2) Data Security and Privacy: Assumption that appropriate security measures will be implemented to protect user data and ensure privacy. 3 ) Scalability: The project assumes that the platform can be scaled up as the user base grows. 4) User Engagement: Assumption that users will actively engage with the platform and populate it with content.

## Dependencies

Resource Availability: The availability of required resources, including human resources and technology infrastructure, is assumed for project execution. Data Security and Privacy: Assumption that appropriate security measures will be implemented to protect user data and ensure privacy. Scalability: The project assumes that the platform can be scaled up as the user base grows. User Engagement: Assumption that users will actively engage with the platform and populate it with content.

# **SYSTEM ARCHITECTURE**

## The system architecture for the LinkedIn Clone project is designed to ensure scalability, maintainability, and security while delivering the core features and functionalities of a professional networking platform. Here's a high-level overview of the system architecture:

## Client-Side:

## Web Application: The client-side of the LinkedIn Clone is a web-based application accessible through web browsers on various devices, including desktop and mobile. It utilizes modern web technologies such as HTML5, CSS3, and JavaScript, with a responsive design for optimal user experience.

## User Interface (UI): The UI is designed to be intuitive and user-friendly, offering features for user registration, profile management, networking, content creation, job search, and more. Users interact with the platform through a visually appealing and customizable interface.

## Front-End Framework: The front-end of the application is built using a JavaScript framework like React.js, ensuring efficient rendering, interactivity, and seamless user interactions. Redux or a similar state management library is used for managing application state.

## Application Server:

## Web Server: A web server (e.g., Nginx, Apache) acts as a reverse proxy and serves the static assets (HTML, CSS, JavaScript) to clients. It also manages SSL/TLS encryption for secure communication.

## Architectural Strategies

## Load Balancing: Implement load balancing to evenly distribute incoming traffic across multiple application server instances, improving performance and fault tolerance.

## Caching: Use caching mechanisms to reduce database load and latency, improving response times for frequently accessed data.

## Security by Design: Integrate security measures at every level of the architecture, including encryption, access controls, and regular security audits, to protect user data and prevent cyber threats.

## Scalability and Elasticity: Design the system to handle increased user loads by horizontally scaling application components and services as needed.

# **DETAILED DESCRIPTION OF COMPONENTS**

For detailed description of the components, please refer **Appendix A – Detailed Description of Components**

The below template will be used to specify the details of all the components

**Table 1: Detailed Design Specification Template**

|  |  |
| --- | --- |
| **Identification** | The unique name for the component and the location of the component in the system. |
| **Type** | A module, a subprogram, a form, a data file, a control procedure, a class, etc. |
| **Purpose** | Function and performance requirements implemented by the design component, including derived requirements. Derived requirements are not explicitly stated in the SRS - but are implied or adjunct to formally stated SDS requirements. |
| **Subordinates** | The internal structure of the component, the constituents of the component, and the functional requirements satisfied by each part. |
| **Dependencies** | How the component’s function and performance relate to other components. How this component is used by other components. The other components that use this component. Interaction details such as timing, interaction conditions (such as order of execution and data sharing), and responsibility for creation, duplication, use, storage, and elimination of components. |
| **Interfaces** | Detailed description of all external or internal interfaces as well as of any mechanism for communicating through messages, parameters, or common data areas. All error messages and error codes should be identified. All screen formats, interactive messages, and other user interface components (originally defined in the SRS) should be given here. |
| **Resources** | A complete description of all resources (hardware or software) external to the component but required to carry out its functions. |
| **Processing** | A full description of the functions presented in the Function subsection. Pseudocode can be used to document algorithms, equations, and logic. |
| **Data** | For the data internal to the component, describes the representation method, initial values, use, semantics, and format. |

# **INTEGRATIONS**

Integrations play a crucial role in expanding the functionality and capabilities of the LinkedIn Clone project. Here are some key integrations that can enhance the platform's user experience and features:

**OAuth Authentication**: Integration with OAuth providers (e.g., Google, Facebook) allows users to easily sign up and log in using their existing social media accounts, simplifying the onboarding process.

# **APPENDICES**

## Appendix A – Detailed Description of Components

|  |  |
| --- | --- |
| **Identification** | **Login screen** |
| **Type** | Class/Form/ |
| **Purpose** | This screen insures that only authentic users can create and use the platform |
| **Subordinates** | This contains link to others sources :   * Home page |
| **Dependencies** | The following are the screen linked to this screen   * Home Page |
| **Interfaces** | firebase |
| **Resources** | Used firebase for data storage |
| **Processing** | The required data of user is stored into the firebase authentication tab one can easily delete or add a user |
| **Data** | * The data for the Login Screen is the username and password entered by the user. * It is validated with a query against the database. |

|  |  |
| --- | --- |
| **Identification** | **Home Page** |
| **Type** | Class/HomePage/ |
| **Purpose** | This screen ensures that user can create post also can edit and signout from website |
| **Subordinates** | This contains link to others sources :  Signout, login, notifications etc |
| **Dependencies** | The following are the screen linked to this screen   * Login page |
| **Interfaces** | React styled components |
| **Resources** | Used redux for state managing and also used firebase for data storage of post |
| **Processing** | Tested and analyzed by making various accounts and recognized smooth functionality |
| **Data** | * The data for the Post page is entered by the user. And can be stored dynamically * It is validated with a query against the database. |