Software Requirement Specification (SRS)
for
‘Connecta’
An online social networking system

ANKUSH NIROULA	**(CS23BTKMU11002)**
ANISH MANANDHAR	**(CS23BTKMU11001)**
VIGNAN KOTA	**(CS21BTECH11029)**
DAVID MALOTH	**(CS21BTECH11035)**

Contents

1 Introduction

1.1 Purpose

Connecta is a social networking site that offers an alternative perspective on how we often use these kinds of sites. Users may now easily join, share, and create content on blogs, social networks, wikis, and forums thanks to social media. Due to their extensive reach and attractive features, social network sites (SNSs) have been gaining the interest of corporations, industries, and economists for a long. Millions of people have social profiles on many of the popular platforms including Facebook, Instagram, Snapchat, MySpace, and others. Thousands of SNSs support various interests and behaviors thanks to different technical affordances. While some websites facilitate connections between strangers based on common interests, political opinions, or hobbies, the majority of them promote the upkeep of pre-existing social networks.

1.2 Scope

Connecta is a general social media platform that aims to provide a different experience to the social networking itself. The system aims to have thousands of users in its space who can share information with their circle. The functionality of the system is divided as follows:

In Scope:

- User Registration: Allows the user to register and create their account in the platform.
- Profile Management: Allows the user to manage their profile, which includes managing connections, posts, and updating their information.
- Post Management: Allows the user to create posts and edit their post accordingly.
- Accessibility of the Posts: We incorporate two categories of posts according to who can view the post; public and private. The public posts are visible to everyone, while the private posts are only visible to connections.
- Making Connections: Allows a user to expand their network by sending and accepting connections from different users.
- Searching User accounts; by different tags: Allows the user to search for a particular account based on their username, firstname, lastname, or company affiliation.
- Feed: Social media feed includes the stream of content/posts from different users, showing mostly relevant information to a particular user.

Out of scope:

- Chat
- Suggestions
- Group & Community

1.3 Overview

The document will mostly consist of two parts:

- Overall Description
- Specific Requirements

Overall description describes the major components of the system, assumptions, and dependencies of the system, while specific requirements describe the functions of the system and their roles in the system and the constraints faced by the system.

2 Overall Description

2.1 Product Perspective

Connecta attempts to change the user experience of the social media platform in a way. It provides users with a different interface than the traditional social media platform does and also works in a way so that user can maintain their online profile at a new level.

In addition to making personal connections, this product provides a platform for users to share their ideas, discover new things, and engage in interesting conversations with people they are not even connected with. It also serves as a resource for the public to stay up-to-date on the latest personal or professional achievements of their social circle.

2.2 Product Functions

Connecta will incorporate different functionalities, each being represented by the use cases overviewed below table:

Class of Use Cases	**Use Cases**	**Description of Use Cases**
Use Cases related to user registration and authentication	Create	*Create and verifies user info*
	Register	*Register user info into system*
	Login	*Login user into system*
	Reset password	*Authenticate and change password*
Use Cases related to post management	VIew feed	*Shows post to user*
	Like post	*Like posts from different user*
	Create post	*Create post of their own*
	Delete post	*Delete post of their own*
User Cases related to search functionality	Perform search	*Search other accounts*
	View recent searches	*Shows recent search for the user*
	Delete recent search	*Clear recent search history*
Use Cases related to connection management	View user profile	*Shows profile of other users*

	Send connection request	*Send request to user to connect*
	View connection requests	*View request to user by others to connect*
	Accept connection request	*Accept request sent to user*
	Remove connection	*Delete a connection with another user*
User Cases related to profile management	Edit profile	*Manage and update profile info*
	Delete profile	*Remove a profile*

2.3 User Characteristics

People of all ages who want to create an online profile on any social media website and who are interested in sharing content related to their own slice-of-life to their social circle, can register in the platform. Users can share their posts, comment on other posts, like posts, connect with other users, and maintain their profiles.

2.4 Principal Actors

Connecta, at this stage, only aims to incorporate one general user. Users will be allowed to use the different functionalities offered by the platform.

2.4.1 General Constraints

- Full functionality requires a stable internet connection.
- Optimal performance is achieved on relatively new and updated web browsers.
- Compliance with data privacy regulations ensures user information security.

2.4.2 Assumptions and Dependencies

- The login system depends on Google authentication; changes or updates by Google are promptly addressed.
- Users are assumed to access the platform from devices meeting minimum compatibility requirements.

3 Specific Requirements

3.1 Functional Requirements

We describe the functional requirements using various use cases.

Use case 1: Create
Primary Actor: General user
Pre-Conditions: Internet connection available, Email account available.
Main Scenario:

- The User provides their information. The user info that should be include are:
  - First Name
  - Last Name
  - Email
- Users can register using a Google account.
- The system sends verification to the email address and waits for confirmation.
- User information is registered in the system.
- The system sends the user to create page.

Alternate Scenario:

- Email not confirmed: If not confirmed within 2 minutes, send the user back to the login page and terminate the registration.

Use case 2: Register
Primary Actor: General user
Pre-Conditions: Internet connection available, Email account verified
Main Scenario:

- The user is prompted to create a unique username and sets and confirm password.
- The system registers Username and password for that user
- The system sends the user back to login page.

Alternate Scenario:

- Password and confirm password do not match: Prompt user to reenter.

Use case 3: Login
Primary Actor: General user
Pre-Conditions: Internet connection available, User should be in login paged
Main Scenario:

- The user enters their username/email and password into the respective fields.
- The user submits the login form by clicking the "Login" button.
- The system verifies the entered credentials against the stored user data in the database.
- The system authenticates the user's session and grants access to the application.
- The user is redirected to their dashboard or the application's homepage.

Alternate Scenario:

- If the credentials are incorrect or do not match any existing user account:
    - The system displays an error message indicating that the login attempt failed.
    - The user is prompted to re-enter their credentials or reset their password if necessary.

Use case 4: Reset Password

Primary Actor: General user

Pre-Conditions: User has clicked the forgot password button

Main Scenario:

- The system presents a form or interface for the user to enter the email address associated with the account.
- The user enters their email address into the provided field.
- The user submits the form to request a password reset.
- The system verifies the email address provided by the user.
- The system sends an email containing the password reset link to the user's email address.
- The user checks their email inbox and clicks on the password reset link.
- The user is redirected to a page where they can create a new password.
- The user enters their new password into the provided fields.
- The user submits the form to confirm the password reset.
- The system updates the user's password in the database.
- The system confirms the successful password reset to the user.
- The user is redirected to the login page

Alternate Scenario:

- If the email address provided is not registered, the system informs the user that it is not recognized and prompts them to retry with a valid email address.

Use Case 5: View Feed

Primary Actor: User

Preconditions: User is logged into the system

Main Scenario:

- The system retrieves a list of posts from the user's network or community.
- The system displays the posts in the feed, sorted by most recent.
- The user can view the post posted by the immediate connections, which is privately/public accessible
- The user can view the posts posted by the second-degree connections which have been made publicly accessible
- The user can like the post
- The system can show the comments
- For each post displayed, the user is presented with several options:
    a. Like: The user can express appreciation for a post by liking it.
    b. Comment: The user can engage in conversation by commenting on the post.

Alternate Scenario:

- If the user has no connections, the system displays the user to send connections to different accounts.

Use Case 6: Like Post/Comments

Actor: User

Preconditions: User logged into their account, User has accessed the feed and viewed a specific post.

Main scenario:

- The user navigates to the post they wish to like.
- The system presents a "Like" button associated with the post.
- The user clicks on the "Like" button.
- The system increments the like count for the post by one.
- The system records the user's action of liking the post.
- The like action is visually reflected to the user, indicating that they have successfully liked the post.
- If the user has already liked the post, the system can unlike the post.

Alternate Scenario :

- If there are technical issues or errors during the like process, the system provides appropriate error messages and prompts the user to retry the action.

Use Case 7: Create Post

Actor: User

Preconditions: User has accessed the posting interface.

Main Scenario:

- The user navigates to the posting interface.
- The system presents a form for creating a new post.
- The user enters the content for the post.
- The user clicks the "Post" button to submit their post.
- If desired, the user attaches media files, such as images or videos, to the post.
- The system publishes the post to the user's profile or feed, making it either private or public
- The system validates the post content, ensuring it meets any length or format requirements.
- The system stores the post in the database, associating it with the user's account.
- If the user decides not to create a post, the user can cancel the post creation process, returning to the previous interface.

Alternate Scenario:

- If the user wants to cancel the process, the system redirects it to their previous interface where they were in.

Use Case 8: Delete Post

Actor: User

Preconditions: User has authored the post they wish to delete, User has accessed the feed and is viewing the specific post they want to delete.

Main Scenario:

- The user navigates to the post they wish to delete.
- The system presents a "Delete" option or button associated with the post.
- The user clicks on the "Delete" option.
- The system prompts the user to confirm the deletion.
- The user confirms their intent to delete the post.
- The system permanently removes the post from the system's database.
- Any associated likes, comments, and replies are deleted from the system to maintain data integrity.

Alternate Scenario:

- If there are technical issues or errors during the deletion process; the system provides appropriate error messages

Use Case 9: Perform Search

Actor: User

Preconditions: User is logged into their account, Search functionality is accessible within the user interface.

Main Scenario:

- The system presents a search bar or interface for entering keywords.
- The user can enter their keyword (Username, First Name, Last Name, Company Name)into the search bar.
- The user submits the keyword by pressing the "Search" button or hitting Enter.
- The system processes the keyword and retrieves relevant results from the database.
- The system displays the search results to the user, typically in a list format.
- The user can view individual profiles coming up from search results.

Alternate Scenario:

- If the keyword does not return any results, the system notifies the user that no results were found for their query.

Use Case 10: Recent Searches

Actor: User

Preconditions: User has accessed the search bar, User has previously searched for some account

Main Scenario:

- The system retrieves the list of recent search queries associated with the user's account.
- The system displays the list of recent searches to the user, typically in chronological order.
- The user can view the recent search queries listed on the screen.
- Optionally, the user may click on a recent search query to perform the search again.

Alternate Scenario:

• If no recent searches are available, the system informs the user that there are no recent searches to display.

Use Case 11: Delete Recent Search
Actor: User
Preconditions: User has made some recent searches
Main Scenario:

- The user accesses the search feature.
- The system retrieves the list of recent search queries associated with the user's account.
- The user selects the specific recent search query they wish to delete from the list.
- The user initiates the deletion action, typically through a delete button on the side.
- The system prompts the user to confirm their intent to delete the selected recent search query.
- The user confirms the deletion.
- The system removes the selected recent search query from the list of recent searches associated with the user's account.

Alternate Scenario:

• If no recent searches are available, the system informs the user that there are no recent searches to delete.

Use Case 12: View User Profile
Actor: User
Preconditions: User has navigated to the profile section of other accounts.
Main Scenario:

- The user navigates to the profile viewing section within the application.
- The system retrieves the profile information associated with the account.
- The system displays the user's profile information, including profile icon, username, bio (if any), posts or content shared by user.
- The user can browse through different profile sections,send connect request, view posts and connections count.

Alternate Scenario:

• If user has made connections with the account, then they can view private and public posts. If not, they can view only public posts.

• If there are technical issues or the profile has been deleted, retrieving the user's profile may provide the appropriate error messages to the user.

Use Case 13: Send Connection Request
Actor: User
Preconditions: User is logged into their account, User is viewing the profile of another user whom they want to connect with.
Main Scenario:

- The user navigates to the profile of another user whom they want to connect with.
- The system displays options to interact with the user's profile.
- The user selects the option to send a connection request to the other user.
- The system sends a connection request to the other user.
- The system updates the status of the connection request in both users' accounts as pending.
- The other user receives a notification of the incoming connection request.

Alternate Scenario:

- If the user has already sent a connection request to the account, the system shows that they have already sent a connection request.

Use Case 14: View Connection Requests

Actor: User

Preconditions: User is logged into their account

Main Scenario:

- The user navigates to the application section dedicated to managing connection requests.
- The system retrieves the list of pending connection requests associated with the user's account.
- The user can view accounts that initiated the connection request, including the sender's profile information.
- The user has the option to accept or reject each connection request.

Alternate Scenario:

- If the user has no pending connection requests, the system informs the user that there are no pending requests to be displayed.

Use Case 15: Accept Connection Request

Actor: User

Preconditions: User is logged into their account and user has received a connection request from another user.

Main Scenario:

- The user receives a notification or accesses the application section for managing connection requests.
- The system retrieves the list of pending connection requests associated with the user's account.
- The user selects the specific connection request they wish to accept from the list.
- The user confirms their intent to accept the connection request.
- The system updates the status of the connection request to "accepted" in both users' accounts.
- The system adds the sender of the connection request to the user's list of connections.

Alternate Scenario:

- If the user has no pending connection requests, the system informs the user that there are no pending requests to be displayed.
- If the user decides to reject the connection request, the request pending is removed from the section, and the sender’s list of pending requests is also updated.

Use Case 16: Remove connection

Actor: User

Preconditions: User is logged into their account, User has made connection with another user.

Main Scenario:

- The user navigates to the profile of another user whom they want to remove connection with.
- The user initiates the remove connection through a button interface.
- The system prompts the user to confirm removing connection.
- When the user confirms the action, the other account is removed from the connection list of the user, and the private posts can not be viewed by one another.

Use Case 17: Edit Profile

Actor: User

Preconditions: User is logged into their account.

Main Scenario:

- The user navigates to the profile editing section of the application.
- The system retrieves the user's current profile information from the database.
- The user modifies their profile information, such as:
    - Updating profile picture
    - Editing bio or description
    - Changing other profile information
    - Adjusting privacy settings
- The user saves the changes by clicking an "Update" button.
- The user saves the changes by clicking an "Update" button.
- The system updates the user's profile information in the database.
- The system displays a confirmation message indicating that the changes have been saved successfully

Alternate Scenario:

- • If the user cancels the deletion process, the system returns to the profile view mode.
- • If the user tries to change the username, the system will not let the user do it.

Use Case 18: Delete Profile

Actor: User

Preconditions: User must have an account and must be logged in.

Main Scenario:

- The user navigates to the profile management section where they intend to delete the profile.
- The system presents a confirmation message informing the user about the consequences of deleting their profile, including the permanent loss of all data associated with the account.
- The user confirms their intent to delete the profile.
- The system initiates the deletion process and permanently removes all user account data from the system's database
- The system removes connections with other accounts that have been linked to the user as well.

- The system logs out the user and redirects them to the application's login page.

Alternate Scenario:

• If the user cancels the deletion process, the system returns to the profile view mode.

3.2 Performance Requirements

Connecta must be interactive, and the delay involved must be minimal. So, in every action response of the software, there is no immediate delay. We aim to focus on two of the following requirements for our software:

- The system should run seamlessly in modern and updated web browser environments.
- When searching for other users, latency should be minimal.

3.3 Design Constraints

The design constraints of Connecta include:

- **Safety:**
  Posts and user data should be protected from viruses and other malicious manipulations in a database.

- **Fault Tolerance:**
  Posts and user data should be safe in a database that can handle power outages and system crashes.

3.4 External Interface Requirements

Connecta aims to incorporate different user interfaces than the existing social media systems available in the market.

- User-friendly interfaces
- API support for potential integrations with third parties

4 Future Extensions

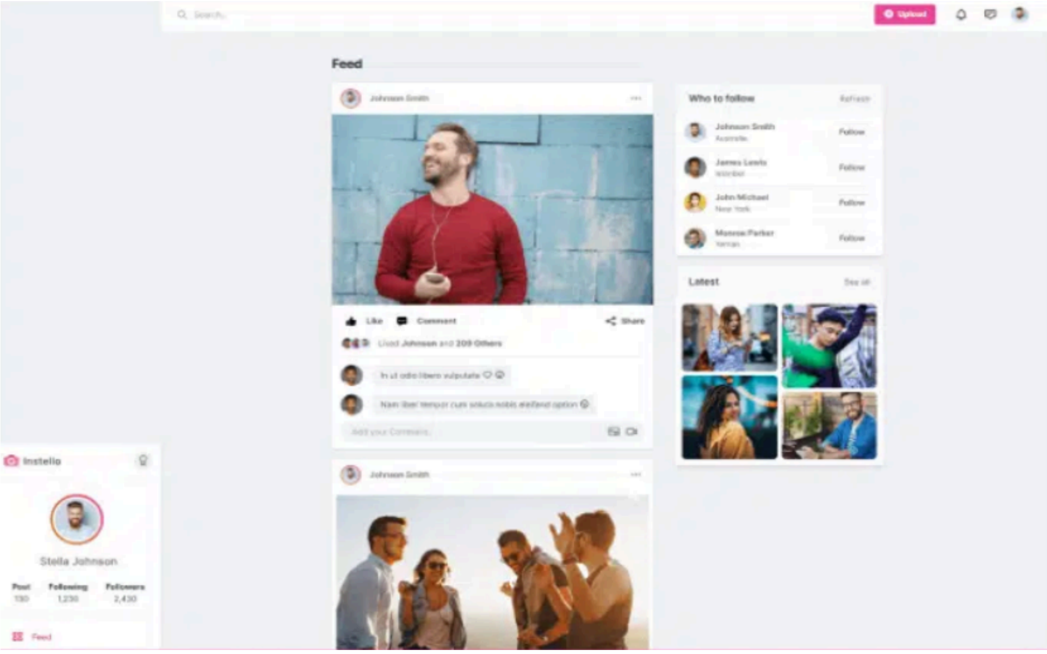
In order to enhance user experience and further improve platform functionality, the following features are under consideration:

- Implement a robust chat management system to fulfill the direct message requirements
- Integrating a machine learning based model to provide user connection suggestions.

5 Appendix.

5.1 Appendix A: The User Screen

This section below shows how the feed will look for the main user.



In ut odio libero vulputate

Nam liber tempor cum soluta nobis eleifend option

Fig 1: user feed screen

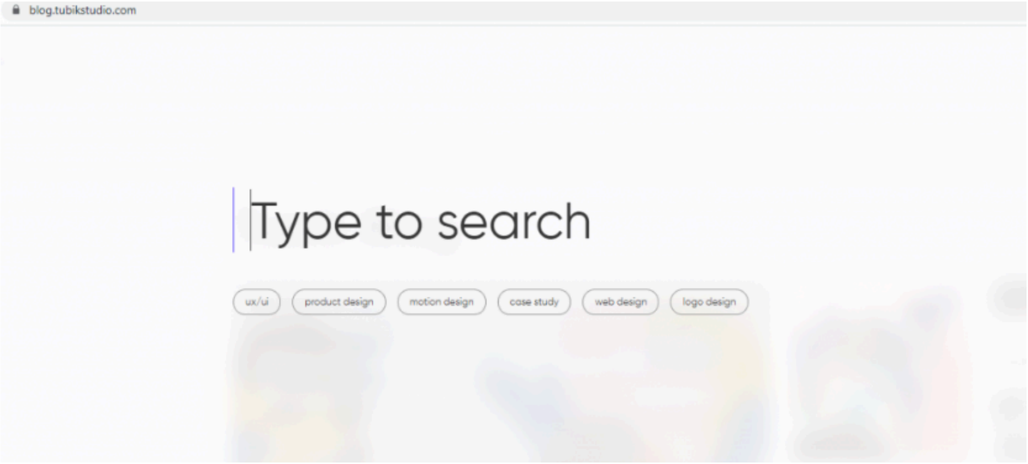
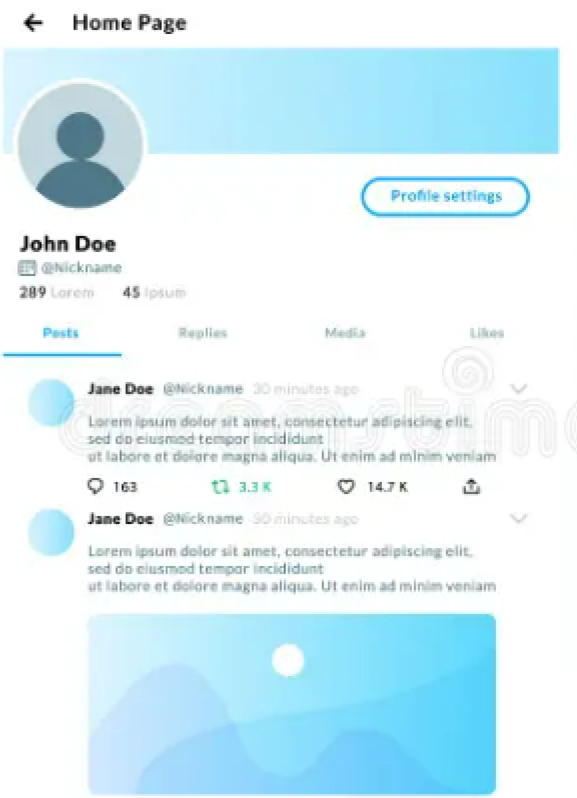


Fig 2: Search screen



Jane Doe @Nickname 30 minutes ago

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad minim veniam

Jane Doe @Nickname 30 minutes ago

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Ut enim ad minim veniam

Fig 3: Profile screen