Connecta

An online social networking system

Introduction

- Connecta is an online social networking system(SNS), designed to connect people from different backgrounds.
- It provides a platform for individuals to build and maintain connections
- The system aims to have thousands of users in its space who can share information with their circle
- Connecta is a social networking site that offers an alternative perspective on how we often use these kinds of sites
- 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.

Key Features

- User Registration:

  Allows the user to register and create their account in the platform.

- Feed:

  Social media feed includes the stream of content/posts from different users, showing mostly relevant information to a particular user.

- 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, delete and edit their posts accordingly.

Key Features(Contd.)

- Accessibility of the Posts:

  - Posts are divided into two categories: public and private, based on 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 or Connection Management:

  Allows a user to expand their network by sending and accepting connections from different users.

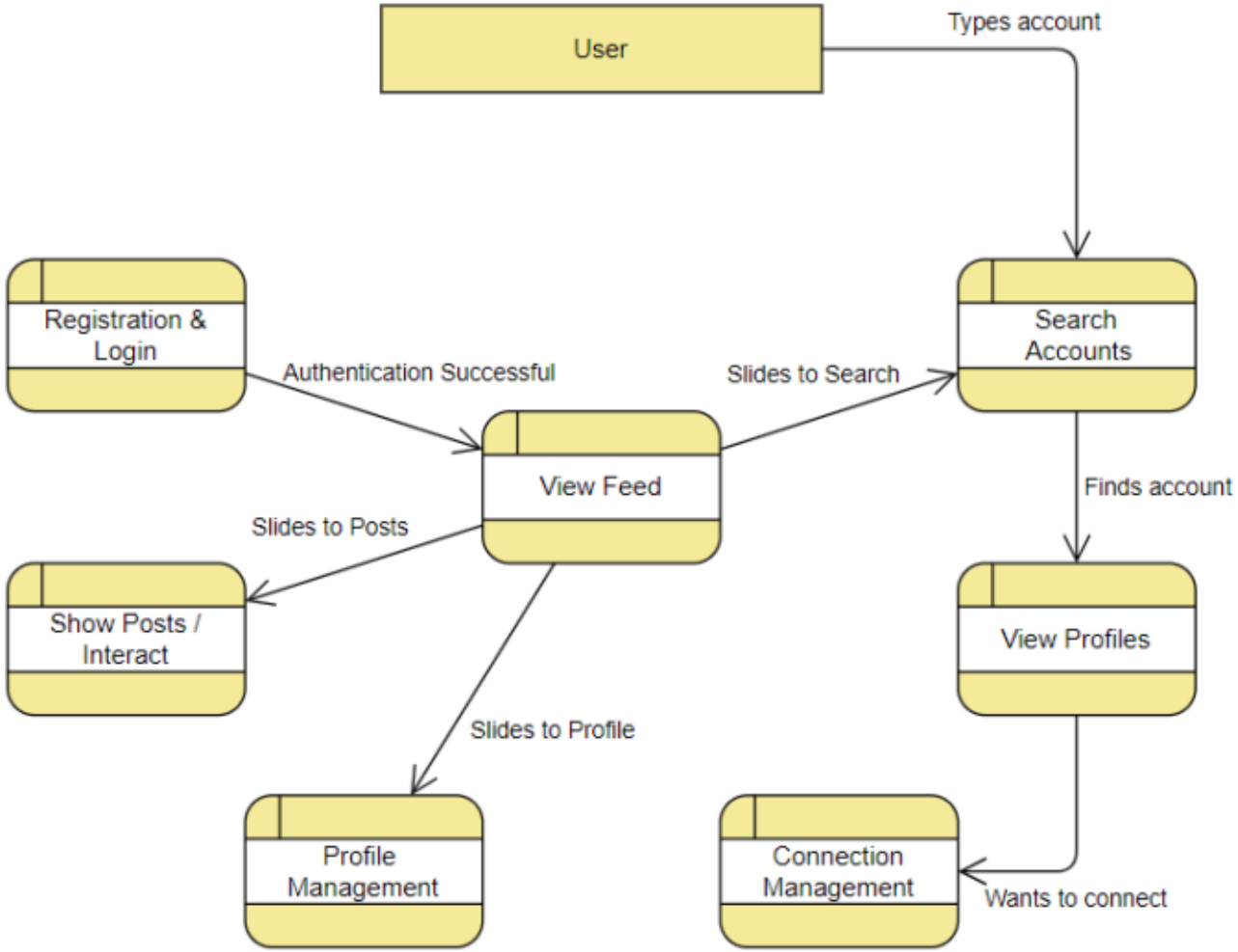
- Searching User accounts by different tags or Search Management:

  Allows the user to search for a particular account based on their username, first name, last name, or company affiliation.

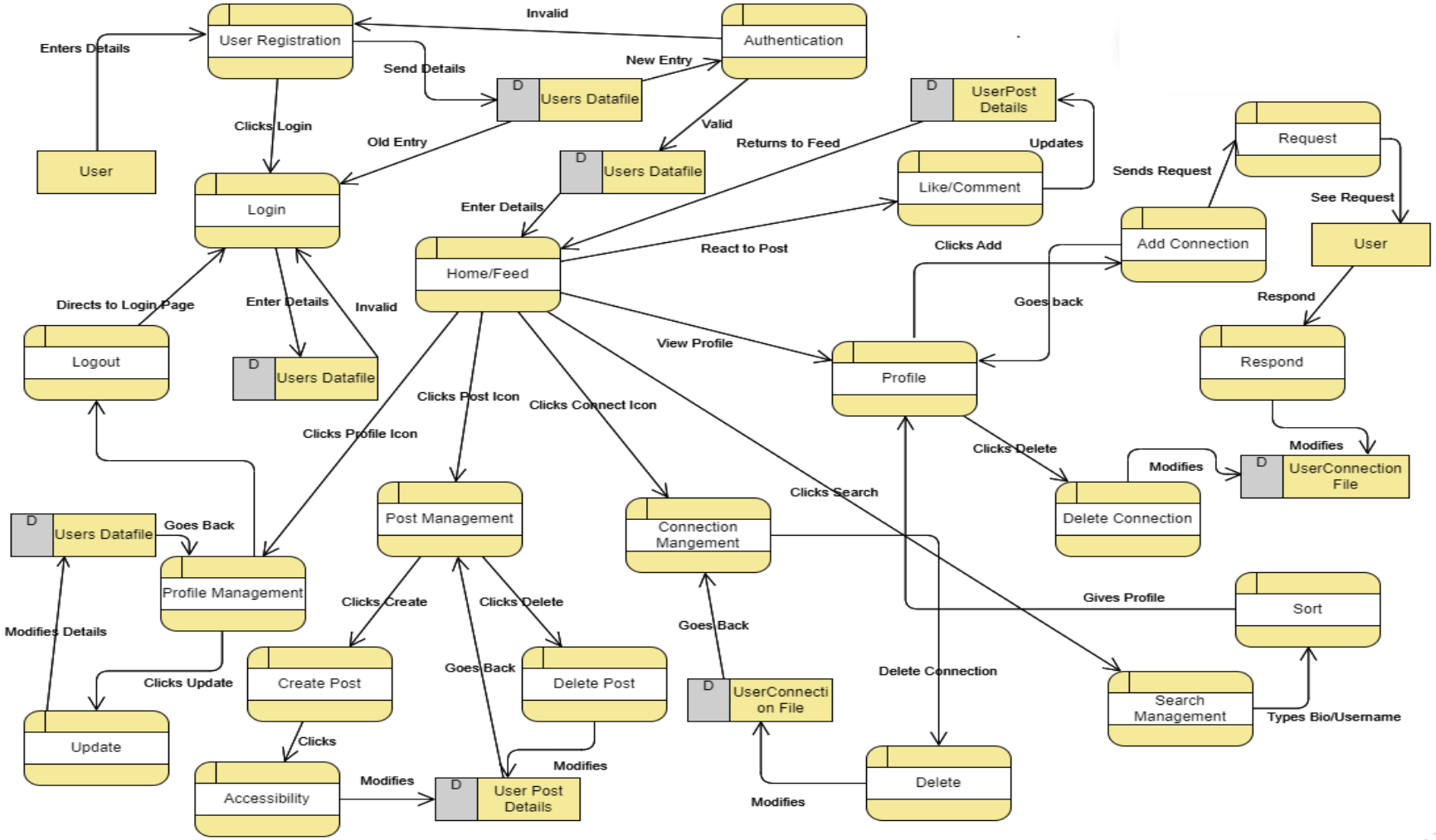
Class of Use Cases	Use Cases	Description of Use Cases
Use Cases related to user registration and authentication	Create	Create and verifies user info
	Register	Register user info into system
	Login	Login user into system
	Reset password	Authenticate and change password
Use Cases related to post management	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

Use Cases

DFD 1


User
Types account
Registration & Login
Authentication Successful
Slides to Search
Search Accounts
View Feed
Finds account
Slides to Posts
Show Posts / Interact
View Profiles
Slides to Profile
Profile Management
Connection Management
Wants to connect

DFD 2


User Registration
Authentication
Users Datafile
UserPost Details
User
Login
Like/Comment
Request
Add Connection
User
Home/Feed
Logout
Profile
Respond
UserConnection File
Delete Connection
Post Management
Connection Mangement
Profile Management
Sort
Create Post
Delete Post
UserConnecti on File
Search Management
Update
Accessibility
User Post Details
Delete
Invalid
Enters Details
Send Details
New Entry
Clicks Login
Old Entry
Valid
Returns to Feed
Updates
Sends Request
See Request
Enter Details
React to Post
Clicks Add
Directs to Login Page
Enter Details
Invalid
Goes back
Respond
View Profile
Clicks Post Icon
Clicks Connect Icon
Clicks Profile Icon
Clicks Delete
Modifies
Modifies
Clicks Search
Goes Back
Gives Profile
Modifies Details
Clicks Create
Clicks Delete
Goes Back
Delete Connection
Clicks Update
Goes Back
Types Bio/Username
Clicks
Modifies
Modifies
Modifies

Reasons for selecting DFD 2:

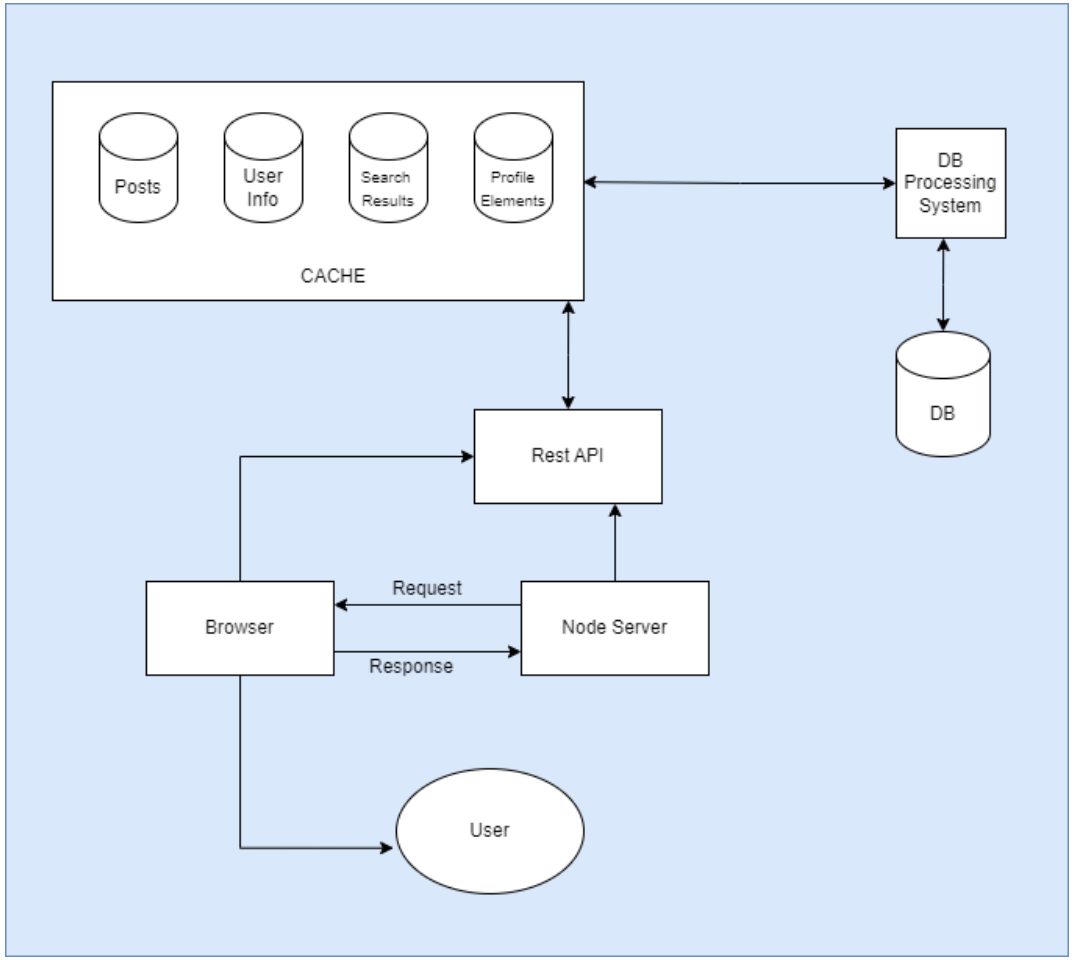
- Level-2 DFD gives detailed insights into processes in social networking.
- Explicitly shows data flows and transformations in each sub-process.
- Crucial for understanding how user data is processed, stored, and shared.
- Enables identification of areas for optimization, process refinement, and enhancing data security.
- Better choice for a thorough understanding of the online social networking system.

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.

Software Architecture

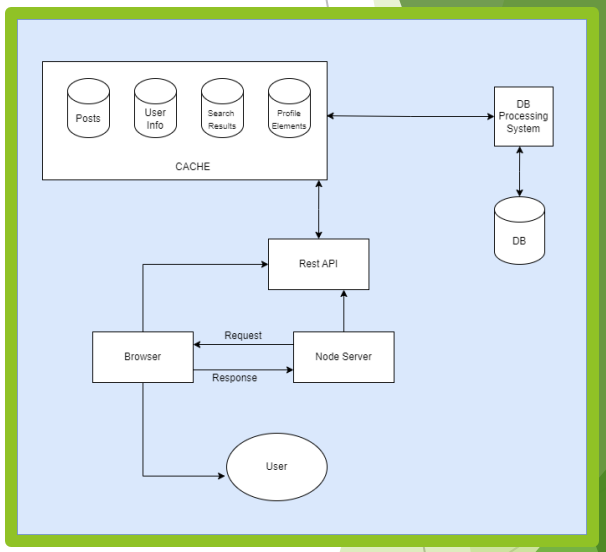
- Architecture 1: Shared Data Style Architecture with Cached Repository



Software Architecture

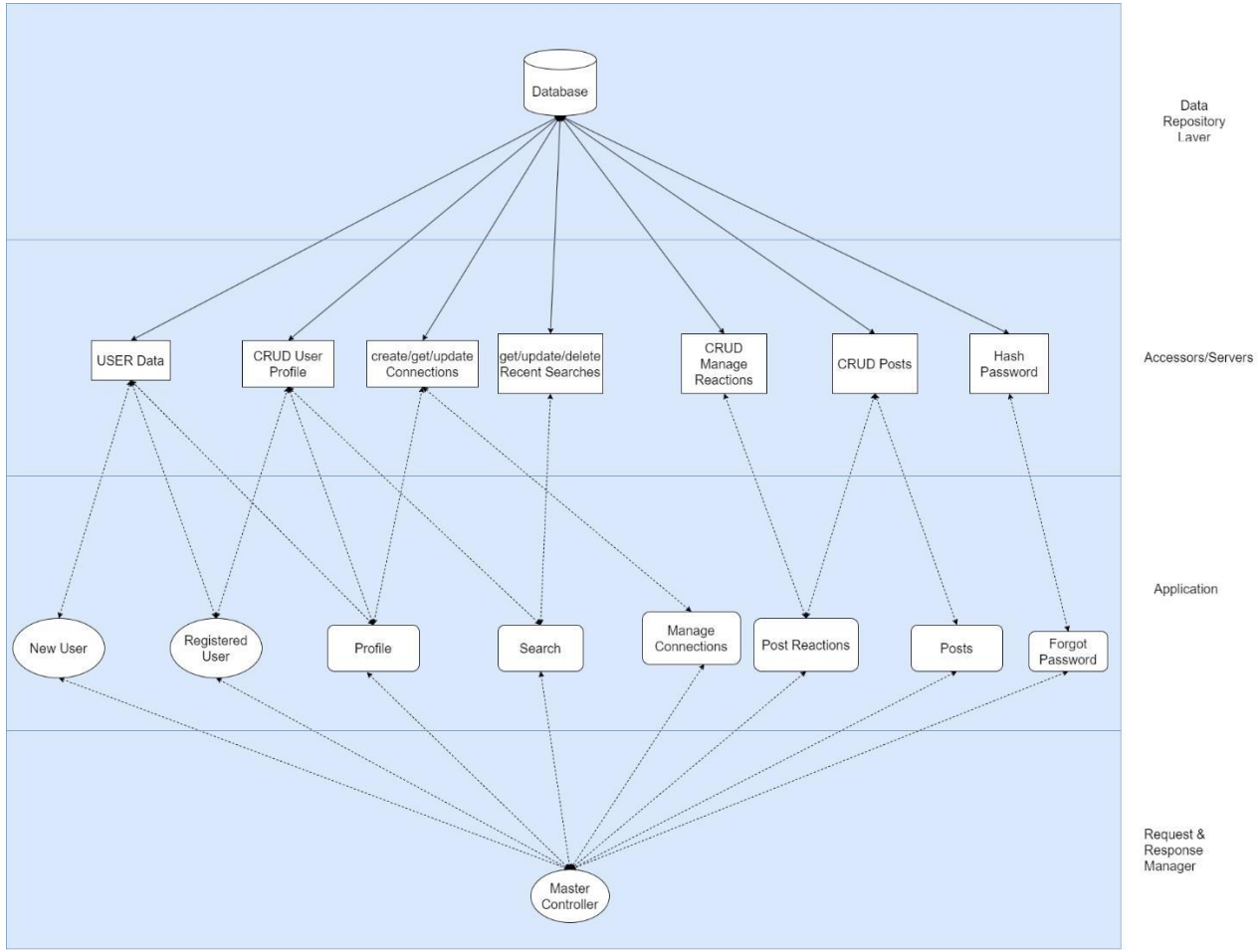
- Architecture 1: Shared Data Style Architecture with Cached Repository (contd.)

- Components are organized for easy data sharing and Browser serves as the main user interface
- User Interaction
  - When a user takes an action (e.g., views a webpage or submits a form), the browser sends an HTTP request to the server.
  - The API processes the request, retrieves necessary data, and uses frontend technologies (HTML, CSS, JavaScript) to create the client-side view.
  - The rendered view is included in an HTTP response sent back to the browser.
  - The interaction cycle concludes when the browser displays the received content to the user.



Software Architecture

- Architecture 2: Shared Data Style Architecture for Connector and Component View (4-Layered model with blackboard variation)

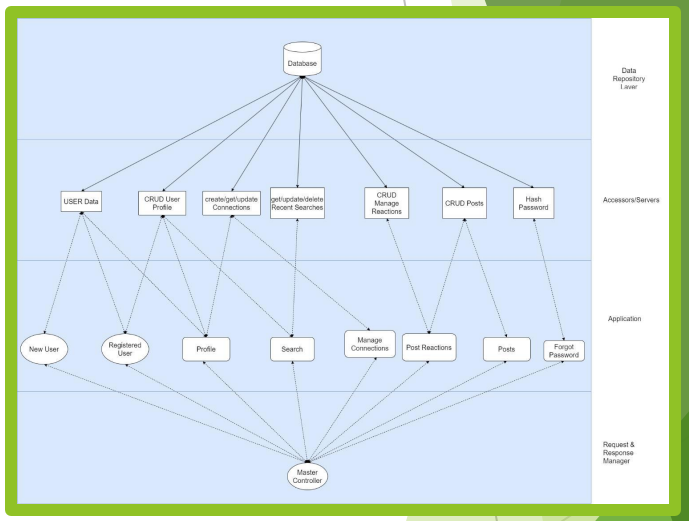


Software Architecture

- Architecture 2: Shared Data Style Architecture for Connector and Component View (4-Layered model with blackboard variation)

The topmost layer features an active data repository (blackboard) that facilitates real-time data sharing and updates. Real-time data availability enhances responsiveness and reduces latency.

The middle layer consists of data accessors that can update or retrieve data from the blackboard. Multiple components can simultaneously read from and write to the blackboard, promoting concurrency and reducing bottlenecks.

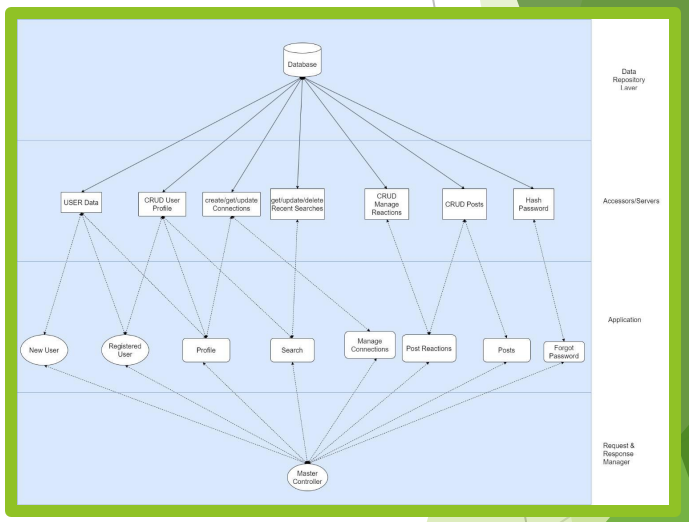


Software Architecture

- Architecture 2: Shared Data Style Architecture for Connector and Component View (4-Layered model with blackboard variation)

The third layer contains application components that interact with the server or data accessors. Swift data retrieval and updates are crucial for a social media platform like Connecta, where real-time interactions are frequent.

The final layer houses a master controller that acts as a request and response manager. Centralized management streamlines communication between users and application services.



Scenarios of interest

S1 : User Registration and Profile Creation: A new user registers to the Connecta and creates a profile and posts text, images to the profile. (Normal Load, all servers are up and database is up)

S2 : Any data accessors or database is down

S3 : Highly loaded system.

	Architecture 1	Architecture 2
Security(S1)	No	Yes
Response Time (S1)	Higher	Lower
Response Time (S2)	High	N/A
Response Time (S3)	Higher	Lower
Data Currency (S1)	Yes	Yes
Debugging	Difficult	Easier
Creating New Features	Relatively Difficult	Easier

Non Functional Attributes

Security

- Robust user authentication
- Data protection
- Single sign on

Privacy

- Manage who can see one's content
- Hash password before storing them

Non Functional Attributes (contd.)

Performance

- Different data accessors help in decreasing response time, as the architecture portrays.

Reliability

- The servers/accessors will use redundant component to handle failures
- Can implement rollback mechanism to ensure data consistency

Non Functional Attributes (contd.)

Modifiability

- The shared-data model lets us change individual components by keeping data producers and consumers apart
- Changes to one component won't directly impact others.

Future Extensions

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.

Conclusion

- By doing Function Point analysis, we got 131 Adjusted Function Points for the system.
- For JavaScript, based on industry data, the average (LOC/AFP ratio) is 64 LOC/AFP
- So, the estimated size of code in JavaScript would be approximately 8,384 lines

Group 04

ANISH MANANDHAR - CS23BTKMU11001

ANKUSH NIROULA - CS23BTKMU11002

VIGNAN KOTA - CS21BTECH11029

DAVID MALOTH - CS21BTECH11035

Thank you