Workshop No. 2 Data Base II

Universidad Distrital Francisco José de Caldas
Computer Engineering
Cuellar B. Paola A.
Ebratt S. Johan D.

1 Architecture Diagram

The following diagram presents the architecture of the social network currently under development. This diagram illustrates the main components of the system, as well as the interactions between them, providing an overview of its structure and functionality.

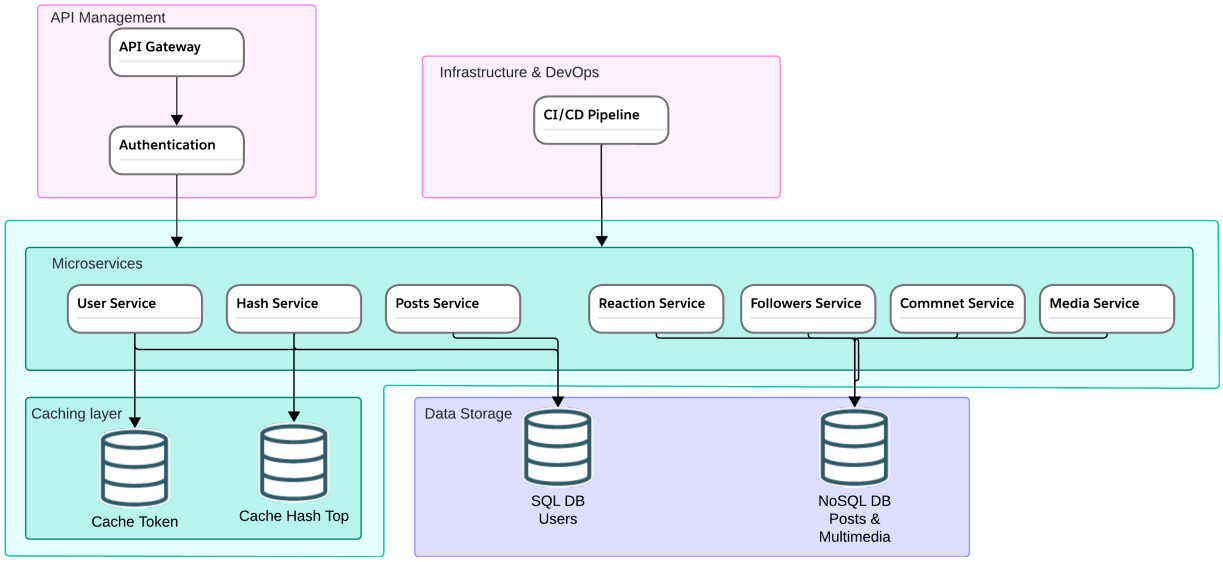

Figure 1: Architecture Diagram

2 Information Requirements

The following section provides a list of the information related to the system requirements. This compilation details the essential functionalities and constraints identified during the analysis phase.

Requirement	**Detailed Description**	**Target Users**	**Related UH**
Personalized post feed	Retrieve recent posts from followed users, prioritized by relevance, date, or interactions.	Members	UH-1, UH-2, UH-11
User profile details	Display name, profile picture, biography, linked social accounts, and activity stats.	Members, companies	UH-4, UH-5
Posts by user	List all posts made by a specific user, including date, content, multimedia, and reactions.	Members, analysts	UH-1, UH-2, UH-6
Post comments	Retrieve comments on a post in chronological order.	All users	UH-11
Reactions per post	Obtain the number and type of reactions (likes, hearts, etc.) for a specific post.	All users	UH-10
Followers and following	Retrieve follower/following lists, with filters by name or user type.	Members	UH-2
User and post search	Search users by name or nickname, and posts by keyword, hashtag, or topic.	All users	UH-12
Reported content	Show details of reported users or posts for review by administrators.	Administrators	UH-15, UH-16, UH-7
Premium user statistics	Retrieve performance data of posts: views, engagement, follower growth, reach.	Premium members	UH-19
Multimedia linked to posts	List uploaded images/videos with technical details and post associations.	Content creators	UH-12, UH-7
Post edit history	View previous versions of a post with timestamps for auditing or recovery.	Admins, advanced users	UH-6
Recent user activity	Log of recent actions: posts, comments, reactions, new/unfollowed users.	Account owners	UH-1, UH-19
Queries by user type or role	Retrieve filtered user lists by role (person, company, admin) or attributes (verified, influencer, active).	Admins, companies	UH-1, UH-16
Posts by visibility level	List posts by visibility: public, private, followers only, with proper access controls.	Members, admins	UH-6
Promoted or featured posts	Show timeline-prioritized posts due to being boosted or marked as relevant.	All users	UH-19
Hashtag tracking and search	Retrieve the most used hashtags (top N), allow search by hashtags, and show latest posts associated with a given hashtag.	All users, analysts	UH-12, UH-1

3 Query Proposals

This section presents a set of query proposals designed to demonstrate how key information can be retrieved from the system using both SQL and NoSQL approaches. The SQL queries are based on structured relational models, while the NoSQL examples utilize document-oriented paradigms, reflecting the flexibility and scalability of modern database solutions.

3.0.1 SQL Queries

- **Retrieve public user profile**

```
SELECT name, nickname, email, profile_image
FROM user
WHERE id = 101;
```

- **List followers of a user**

```
SELECT u.nickname
FROM follower f
JOIN user u ON f.fk_follower_user = u.id
WHERE f.fk_followed_user = 101;
```

- **Top users by number of posts**

```
SELECT u.nickname, COUNT(p.id) AS post_count
FROM user u
JOIN post p ON p.fk_user_id = u.id
GROUP BY u.nickname
ORDER BY post_count DESC
LIMIT 10;
```

- **Retrieve comments of a post**

```
SELECT u.nickname, c.text_comment, c.date_create
FROM comment c
JOIN user u ON c.fk_user_comment_id = u.id
WHERE c.fk_post_id = 501
ORDER BY c.date_create ASC;
```

- **Reactions per type for a post**

```
SELECT cr.label, COUNT(r.id) AS total
FROM reaction r
JOIN cat_reaction cr ON r.fk_reaction_id = cr.id
WHERE r.fk_post_id = 501
GROUP BY cr.label;
```

- **Recent posts by a user**

```
SELECT text_post, date_post
FROM post
WHERE fk_user_id = 101
ORDER BY date_post DESC
LIMIT 5;
```

- **Posts with attached multimedia**

```
SELECT p.id, p.text_post, a.name AS file_name, a.format
FROM post p
JOIN attached_multimedia a ON a.id = p.id;
```

- **Edit history of a post**

```
SELECT hp.text_version, hp.date_modificate
FROM historical_post hp
WHERE hp.id = 501;
```

- **Top users by number of reactions**

```
SELECT u.nickname, COUNT(r.id) AS total_reactions
FROM reaction r
JOIN user u ON r.fk_user_react_id = u.id
GROUP BY u.nickname
ORDER BY total_reactions DESC
LIMIT 10;
```

- **Private posts of a user**

```
SELECT id, text_post
FROM post
WHERE fk_user_id = 101 AND visibility_type = 'private';
```

- **Users with most followers**

```
SELECT u.nickname, COUNT(f.fk_follower_user) AS total_followers
FROM user u
JOIN follower f ON u.id = f.fk_followed_user
GROUP BY u.nickname
ORDER BY total_followers DESC
LIMIT 10;
```

- **Most reported users**

```
SELECT u.nickname, COUNT(r.id) AS report_count
FROM report_user r
JOIN user u ON r.fk_report_to_user = u.id
GROUP BY u.nickname
ORDER BY report_count DESC
LIMIT 10;
```

3.0.2 NoSQL Queries

- **Query recent hashtags used (Redis sorted set)**

```
ZREVRANGE hashtags 0 9 WITHSCORES
```

- **Get latest posts by hashtag (Redis list)**

```
LRANGE hashtag:tech:posts 0 9
```