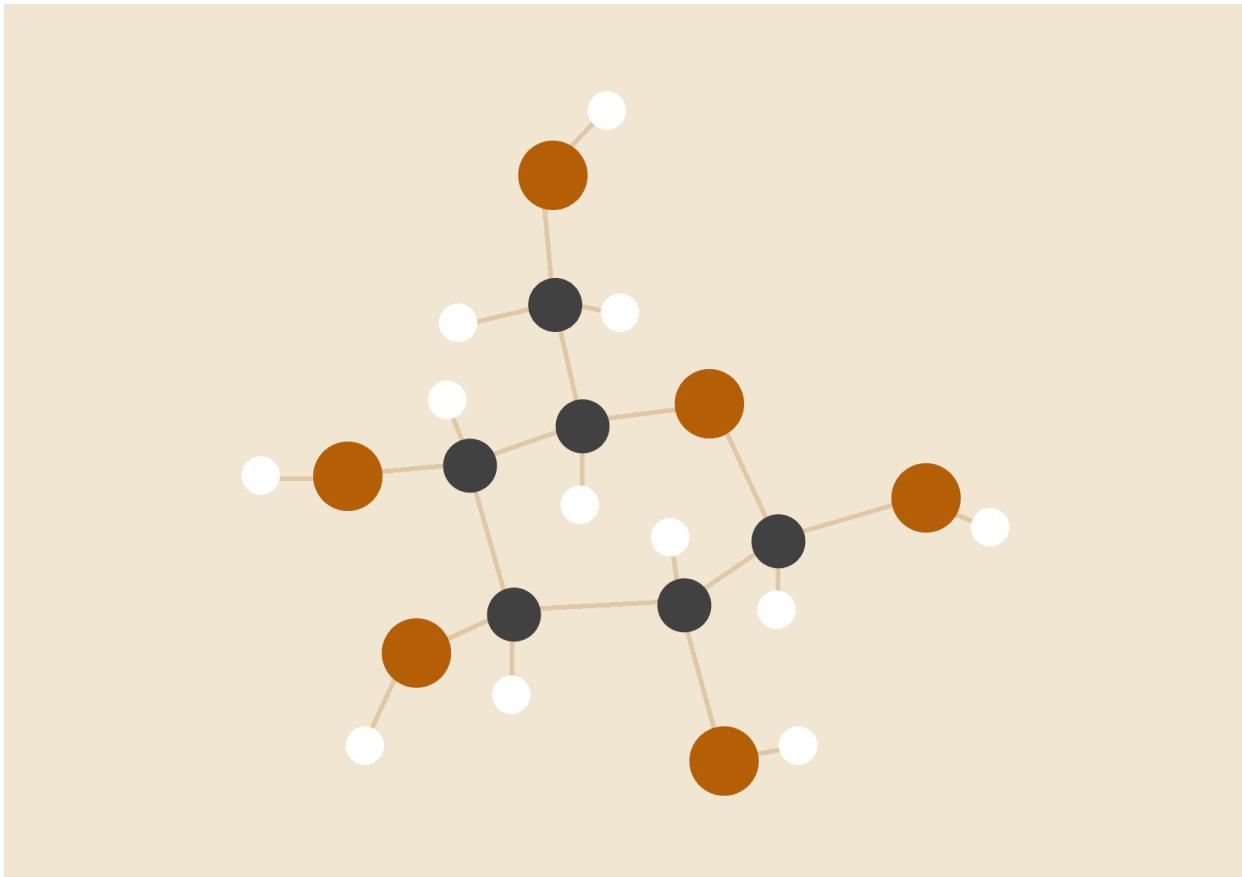


Cluster

Public Collaborative Forum & Social News Aggregation



Mohd Zaid, Nikhil Agrawal & Jai Ahuja

2024353, 2024380 & 2024275

Tutorial Group 1 | Project Group 8

Assigned TAs: Sourav Panja and Shivangi Gaur

INTRODUCTION

Cluster is a web-based platform designed for high-quality, community-driven discussion. It organizes content into "Clusters" (interest groups). Unlike standard social networks, Cluster emphasizes structure and credibility. Features like **Windows** (cluster-references) and **Megaphones** (time-sensitive announcements) allow for sophisticated content curation. The platform prioritizes reputation and moderation, utilizing both human oversight and automated **Bot Rules** to maintain professional standards.

TARGET AUDIENCE

The system utilizes a hierarchical Role-Based Access Control (RBAC) model to ensure data integrity and community quality.

Guest (Unauthenticated)

- Public users or search engine crawlers who have not logged in.
- Strictly Read-Only access. They can browse public Clusters, search for content, and view "Megaphone" announcements.
- They are restricted from voting, commenting, or viewing private communities.

Member (Standard User)

- Registered users who have authenticated via the login system.
- Can submit Posts (Text/Link/Window) and Comments.
- Can use Reactions (Emojis) on content.
- Can manage subscriptions to public Clusters.

Verified (Trusted User)

- A subset of Members who have passed phone/email verification.
- Inherits all Member privileges, plus the exclusive ability to initialize new Clusters. (restriction prevents spam communities).
- Can manage subscriptions to private Clusters.

Moderator (Elevated User)

- Users appointed to oversee specific Clusters.
- Has control of **Management** and **Enforcement** within specific Clusters.

Administrator (Super User)

- Platform maintainers with global access.
- Full database access. They can ban users platform-wide, delete any Cluster, and resolve disputes.

BUSINESS REQUIREMENTS

The following rules define the logical constraints of the system:

1. **Creation of new Clusters** is restricted to **System Admins** and **Verified Members**.
2. A "Window" is a special post type that links to a post in a different Cluster.
 - a. A Window cannot reference a private or deleted post.
 - b. A Window does not fork the discussion.
 - c. A Window can be referenced but only as a Window.
3. Restrictions on Interactions (emojis/stickers).
 - a. Users cannot react (emojis/stickers) to their own content.
 - b. A user can select only **one** Reaction type per post.
 - c. A user can either **Like** or **Dislike** a post/comment/window.
4. Megaphone Constraints and Announcements.
 - a. Only Moderators can mark a post as a "Megaphone" (Announcement).
 - b. Users can subscribe to even countdowns, alarms and other notifications within Clusters (that they have subscribed to).
5. Bot Moderation and Cluster Health.
 - a. Every Cluster can have specific "**Bot Rules**".
 - b. Content violating a Bot Rule must be rejected at the database insertion.

... (to be iterated upon in the future).

FUNCTIONAL REQUIREMENTS

1. **User & Auth:**
 - a. Secure Registration/Login with Phone/Email verification steps.
 - b. Profile management showing user history and "Verified" status.
2. **Content Management:**
 - a. Support for Text, Media, and Windows.
 - b. Recursive comment system allowing infinite nesting (Reply to Reply).
 - c. A set of emoji-based reactions or stickers with counts aggregated per post.
3. **Advanced Features:**
 - a. **Megaphones**
 - b. **View Windows**
 - c. **Bot Configuration**

PRELIMINARY DATA SCOPE

The following entities have been identified for the Conceptual Model:

- **Users:** (ID, Email, Phone, IsVerified, PasswordHash)
- **Clusters:** (ID, Name, Description, CreatorID, IsPrivate)
- **Posts:** (ID, Title, Content, Type [Standard/Window], ClusterID, UserID, Timestamp)
- **Windows:** (WindowID, OriginPostID, TargetClusterID) *[Links a post]*
- **Comments:** (ID, Body, PostID, UserID, ParentCommentID) *[Recursive Relationship]*
- **Reactions:** (UserID, TargetID, ReactionType)
- **Megaphones:** (PostID, StartTime, EndTime, IsActive)
- **BotRules:** (RuleID, ClusterID, RegexPattern, ActionType)

PROPOSED TECH STACK

Backend:

- Primary Service (Elixir & Phoenix Framework)
- Moderation Service, Crawlers and Bot APIs (Python & FastAPI)

Deployment:

- Vercel or netlify

Frontend:

- React Js/Ts

NOTABLE MENTIONS

Whole Group - for deciding to work on **Clusters**.

Mohd Zaid - for ideation of **Window, BotRules & Megaphones**.

Nikhil Agrawal - for **ideation of pipeline** to connect the **backend and frontend**.

Jai Ahuja - for defining the **RBAC model**.