Assignment 1 - Vision and Scope Document

Team 22

September 7, 2025

1 Project Vision

1.1 1.5 Vision Statement

For online communities and niche interest groups who need a fast, minimal, discussion-first platform, our Reddit-style web forum is a social news and threaded-discussion system that provides communities (subforums), posts, comments, voting, ranking (hot/new/top), search and basic moderation. Unlike ad-heavy or bloated alternatives, our product focuses on speed, clarity and a clean API so teams can later build mobile clients and integrations.

1.2 1.4 Success metrics

A measurable criteria that can tell us whether the project is successful and worth undertaking is having 100 new user signups per month for the first 6 months after the launch of the project. And having 3.000 active users by the end of the first year indicating that users prefer our platform over similar platforms. As well as active users having 30 active communities with an average of 10 or more comments and likes (interactions) per week at the end of the first year which shows that users consistently engage with the platform and have a clear preference for this platform compared to competitors.

1.3 2.3 Scope of subsequent releases

The first version is just the basics. Communities, posts, comments, voting, ranking, search and simple moderation. Later releases will add more stuff.

- Advanced Moderation Tools: reporting system and banning so mods can deal with bad users.
- User Experience & Engagement: support for images, videos and links, direct messages, notifications, user profiles with activity and stats. Dark mode (might come with MVP). Also login with Google, GitHub and other providers.
- Community Growth Features: communities can set their own rules, themes and flair. Cross-posting between communities. Event posts and sticky/pinned discussions.

These will come step by step depending on what users want and how the platform grows.

2 Use Case Document

2.1 Title of Fully Dressed Use Case 1 [UC1]

Use case 1: Log into the system

- Scope: Social discussion platform (similar to reddit)
- Level: User goal
- Primary actor: Registered user
- Stakeholders and interests:
    - Registered User: Wants to quickly and securely access their account, communities and personalized feed.
    - Platform Owner (company): Wants to ensure only authorized users log in, prevent account misuse, and collect metrics on active users.
    - System: Needs to authenticate users correctly, ensure system security, and handle failed login attempts gracefully.
- Preconditions:
    - The user must have previously registered and created an account.
    - The system is online and accessible.
- Success Guarantees:
    - The User gains access to their account and the communities they are registered in.
    - If login fails, the system will display an error message.

- Main Success Scenario:
    - The user selects Log in on the home page/app.
    - The system displays the log in page with fields for username/email and password.
    - The user enters valid credentials.
    - The system validates the format of the input fields.
    - The system securely sends the credentials to the authentication server.
    - The authentication server checks if the credentials match a registered account.
    - The system verifies the account is active (not banned, disabled, or deleted).
    - The system redirects the user to their personalized home page.
    - The user is now logged in and can interact with the platform.
- Alternate scenarios:
    - Invalid credentials:
        - User enters a username and password that are invalid.
        - System displays an error message telling the user the username or password is invalid.
        - User can retry and has 5 attempts before the account is temporarily locked.

        - Account inactive (banned, locked, deleted)
            - User tries to log in with a username and password that is either banned, locked deleted.
            - The system displays an error message telling the user that the account is unavailable and to contact support.

- Special requirements:
    - Passwords must be masked while typing them
    - Login page must load within 5 seconds under normal circumstances.

- Frequency of use:
    - Multiple times a day per user.

- Open Issues:
    - How long should an inactive session remain valid until it is logged out automatically?

2.2 Title of Fully Dressed Use Case 2 [UC2] — Create Post

- Scope: Social discussion platform (similar to reddit)
- Level: User goal
- Primary Actor: Authenticated user
- Stakeholders and Interests:
    - Authenticated User: Wants to easily share content (text, link, image) within a chosen community.
    - Community Members: Want relevant, properly formatted, and accessible content.
    - Platform Owner (company): Wants high-quality engagement, enforce posting rules, and ensure reliable storage.
    - System: Needs to validate and persist posts correctly, handle media uploads, and update search/index feeds.
- Preconditions:
    - User is logged in.
    - User has posting rights in the selected community.
- Success Guarantees:
    - Post is successfully saved, indexed, and visible in the community feed.
    - Post is attributed to the correct author.
- Main Success Scenario:
    - User navigates to `/r/{community}/submit`.
    - System displays submission form with fields for title, content type (text/link/image), and body/URL/media.
    - User enters a valid title and selects a post type.
    - System validates inputs (e.g., title length, URL format, media file type/size).
    - System securely uploads media if required.
    - System stores post in the database.
    - System initializes vote score and indexes post for search.

    - System confirms success and navigates the user to the new post's page.
    - Post appears in the "new" feed for the community.
- Alternate Scenarios:
    - Media upload failure:
        - System displays error, user can retry upload or cancel.
    - Disallowed domain for links:
        - System blocks submission and shows a warning.
    - Database transaction failure:
        - System rolls back and shows error, logs the issue internally.
- Special Requirements:
    - Title limited to 300 characters.
    - Images/videos must meet file size and format limits.
    - Submissions must complete within 10 seconds under normal conditions.
- Frequency of Use:
    - Thousands of times per day across the platform.
- Open Issues:
    - Should new posts start at 0 or +1 score?
    - Cross-posting policies between communities.

2.3 Title of Fully Dressed Use Case 3 [UC3] — Vote on Post/Comment

- Scope: Social discussion platform (similar to reddit)
- Level: User goal
- Primary Actor: Authenticated user
- Stakeholders and Interests:
    - Authenticated User: Wants to express approval/disapproval and influence ranking.
    - Content Author: Wants fair voting and visibility of their content.
    - Platform Owner (company): Wants to encourage engagement, prevent abuse/brigading, and maintain accurate ranking.
    - System: Needs to record votes idempotently, recompute scores, and handle concurrent voting.
- Preconditions:
    - User is logged in.
    - Target post or comment exists.
    - User is not banned or rate-limited.
- Success Guarantees:
    - Exactly one authoritative vote per user per target is recorded.
    - Post/comment score and ranking reflect the new vote.
- Main Success Scenario:
    - User clicks upvote or downvote on a post or comment.
    - System records or toggles the user's vote in the database.
    - System recomputes the score and ranking.
    - UI reflects the new vote state and updated score.
- Alternate Scenarios:
    - Concurrent vote submissions:
        - System ensures idempotency; last intent wins.

        - Rate limiting:
            - System returns error, user must wait before retry.
        - Ranking updates delayed:
            - Score updates immediately, ranking is adjusted asynchronously within a few seconds.
- Special Requirements:
    - Vote action must be processed within 1 second under normal conditions.
    - Optimistic UI update required for responsiveness.
- Frequency of Use:
    Extremely frequent; millions of times per day across the platform.
- Open Issues:
    - Should new vote totals be visible immediately, or partially hidden to prevent manipulation?
    - What heuristics should be applied to detect and mitigate vote cheating?

2.4 Brief Use Cases

- Brief Use Case 1 [UC4]: Comment on Post — replies, nesting, edit/delete grace.

User clicks the comment button on a post. A comment field appears. User types his comment and clicks the submit button. The comment appears under the post for others to view.

- Brief Use Case 2 [UC5]: Browse Feeds — hot/new/top; global & per-community.

User opens the feed. By default sees Hot (global). User can switch to New or Top, and choose between global or a specific community. System shows posts with votes and comments. User scrolls to load more and can click into a post.

- Brief Use Case 3 [UC6]: Create/Join/Leave Community — r/{name}, rules, join/leave.

User finds community he wants to join. User clicks Join button. System adds user as a member. User can now participate in posts and interact with posts in the community. To leave the community the user selects leave button on page. This causes the user to be removed from the members list and can no longer see posts within the community or interact with them.

User selects Create community. User gives community name and description. System validates input and creates community. User becomes a member of the new community page.

User is able to leave a Communtiy his is a member of. User give community name and the system validates the input and removes him from the community.

- Brief Use Case 4 [UC7]: View User Profile — posts, comments, karma/score.

User clicks on another user's name. System opens that user's profile page showing their posts, comments, and overall score/karma. User can scroll through activity and click into individual posts or comments.

- Brief Use Case 5 [UC8]: Search Content — posts, communities, users.

User types a query into the search bar. System displays matching communities, posts, and/or users on a results page. User can click a result to navigate to it.

• Brief Use Case 6 [UC9]: Basic Moderation — delete post/comment in own community.
Moderator clicks the Delete button on a post or comment. System removes the content and replaces it with a “Removed by moderator” tag for transparency.

Brief Use Case 7 [UC10]: Register / Sign Up A new user visits the site and selects the Sign Up option. The system displays a registration form with fields for username, email, and password.
The user fills in valid information and submits the form. The system validates inputs (unique username/email, password length and format), securely hashes the password, and creates a new user account in the database.
After successful registration, the user is redirected to the login page or automatically logged in.

• Brief Use Case 8 [UC11]: Log Out / Session Handling

A logged-in user clicks the Log Out button or remains inactive for an extended period.
The system invalidates the active session and removes session data from memory or persistent storage.
The user is redirected to the home or login page and can no longer access protected routes until logging in again.
If the session expires automatically, the next protected request triggers a redirect to the login screen.

• Brief Use Case 9 [UC12]: Edit/Delete Own Content

A user views one of their own posts or comments.
They click the Edit button to update the text; the system displays an editable field and validates the new content before saving.
If they click Delete, the system performs a soft delete (marks as removed) so that other users see a “Deleted by author” placeholder.
Edits and deletions are only available to the content’s original author within allowed time limits or until the content is locked.

Brief Use Case 10 [UC13]: Data Dump — export platform snapshot.

Admin opens the admin/tools page and clicks Export Data. System compiles a full snapshot of users, communities, memberships, posts, comments, and votes. System base64-encodes any post media and returns a single JSON file for download. No database changes occur; the export is read-only and reflects the current state at the time of request.

3 Project Estimation and Prioritization

Lower number a higher priority.

Use Case Priority
UC2 P1
UC1 P1
UC10 P1
UC6 P2
UC8 P2
UC11 P2
UC3 P3
UC4 P4
UC5 P5
UC12 P5
UC9 P6
UC7 P6

4 Project Plan and Schedule

The example schedule for the 10-week project timeline (starting the week after assignment 1 turn-in) is as follows: Note that there are 4 sprints, one for each following assignment. Sprint 1 is two weeks, Sprint 2 and Sprint 3 are three weeks, and the final sprint is two weeks. Decide who is going to be the P.O. for each of the sprints. This is a template for a schedule, adjust as needed.

1
Week Use Cases Expected Hours P.O. (Initials) Sprint Consultation

Week	Use Cases	Expected Hours	P.O. (Initials)	Sprint	Consultation
1	None	XX	Allir	1	A1 Presentation
2	None	XX	SBS	1	Model Drafts
3	None	XX	SBS	2	A2 Presentation
4	UC2, UC1,	12	OVD	2	Dev support
5	UC10	8	OVD	2	Dev support

6	UC6	8	OVD	3	A3 Presentation
7	UC8, UC11, UC3,	10	JJ	3	Dev support
8	UC4, UC5, UC9,	10	JJ	3	Dev support
9	UC7, UC12	6	JJ	3	A4 Presentation
10	Testing		Allir	4	Final presentaton

5 Project skeleton

Every team member has commited, https://github.com/skarihacks/hugbunadarverkefni1
Sævar Breki Snorrason - Sabbalingurinn
Óskar Víkingur Davíðsson - Skarihacks
Jóhannes Jóhannesson - Joij04