

Use Case Descriptions for Real-Time News and Fact-Checking Platform

1. News Verification

1.1 Verify News from External Sources (First-Time Analysis)

Use Case ID: #1-1

Use Case Name: Verify News from External Sources (First-Time Analysis)

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: User, System

Description: Users paste a news post link from social media, and the system verifies its credibility for the first time. A new page is created in the database, containing summaries, verification reports, and a credibility score. The system continuously checks real-time news sources to ensure the most up-to-date verification.

Preconditions: The link has not been analysed before.

Postconditions: A new verification page is created with relevant credibility analysis and community interaction features.

Priority: High

Frequency of Use: High

Flow of Events:

- 1. User pastes a news link into the app.**
- 2. System checks if the link already exists in the database.**
- 3. Since it is a first-time analysis, the system fetches real-time news from an API.**
- 4. System extracts key content from the link and compares it with verified news sources.**
- 5. System generates a summary and a credibility score based on findings.**
- 6. System flags the news as "Verified," "Fake," or "Unverified."**
- 7. System creates a new page in the database for the news link, storing all analysis results.**
- 8. The system continuously monitors real-time news sources for updates and modifies the verification report if new information is available.**
- 9. The user is redirected to this new page.**

Alternative Flows:

- Invalid link: System prompts the user to enter a valid news link.**
- No matching news: System informs the user that no confirmation is available but continues to monitor for updates.**

1.2 Redirect to Existing Verification Page

Use Case ID: #1-2

Use Case Name: Redirect to Existing Verification Page

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: User, System

Description: If the news link has already been verified, the system redirects the user to the existing verification page without reprocessing it. The credibility score and verification details are continuously updated using real-time news sources.

Preconditions: The link has been previously analysed and exists in the database.

Postconditions: The user is redirected to the existing verification page with updated credibility insights.

Priority: High

Frequency of Use: High

Flow of Events:

- 1. User pastes a news link into the app.**
- 2. System checks if the link already exists in the database.**
- 3. Since it has been analyzed before, the system retrieves the stored verification page.**
- 4. System fetches new information from real-time news sources to ensure up-to-date verification.**
- 5. System updates the credibility score and verification status if new data is available.**
- 6. Users are redirected to the verification page, where they can view AI analysis, real-time updates, and community notes.**

Alternative Flows:

- Expired data:** If the stored analysis is outdated, the system re-runs verification before redirecting.

2. Fact-Checking and Community Notes

2.1 Add a Community Note

Use Case ID: #2-1

Use Case Name: Add a Community Note

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: Registered User

Description: Users can add fact-checking notes to news articles, similar to Twitter Community Notes.

Preconditions:

- **User is logged in.**

Postconditions:

- **The note is stored and displayed with the article after approval.**

Priority: High

Frequency of Use: Medium

Flow of Events:

- 1. User selects an article and clicks "Add Community Note."**
- 2. User enters the note with supporting sources.**
- 3. System validates the input and sends it for review.**
- 4. The note is either approved or rejected based on community and moderator review.**
- 5. Approved notes are displayed under the article with credibility ratings.**

Alternative Flows:

- **Invalid input: If the note lacks sources or is empty, the system prompts for revision.**
- **Pending review: Notes require peer approval before being displayed.**

2.2 Add a Community Note When None Exists

Use Case ID: #2-2

Use Case Name: Add a Community Note When None Exists

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: Registered User

Description: If an article has no existing community notes, a user can add the first note to initiate fact-checking discussions.

Preconditions:

- User is logged in.
- The article has no existing community notes.

Postconditions:

- The note is stored in the system and awaits approval.
- The note appears under the article after approval.

Priority: High

Frequency of Use: Medium

Flow of Events:

1. User selects an article and sees that no community notes exist.
2. User clicks "Add Community Note."
3. User enters a note with supporting sources.
4. System validates the input and sends it for review.
5. The note is approved or rejected based on community and moderator review.
6. If approved, the note is displayed under the article.

Alternative Flows:

- Invalid input: System prompts the user to add proper sources or revise the note.
- Pending review: The note remains hidden until approved.

2.3 Upvote Community Notes

Use Case ID: #2-3

Use Case Name: Upvote Community Notes

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: Registered User

Description: Users can upvote existing community notes to indicate their credibility and usefulness.

Preconditions:

- **User is logged in.**
- **The article has at least one community note.**

Postconditions:

- **The note receives an upvote from the user.**
- **The system updates the note's ranking based on total upvotes.**

Priority: Medium

Frequency of Use: High

Flow of Events:

- 1. User selects an article with existing community notes.**
- 2. User reviews the available notes.**
- 3. User clicks the upvote button on a note they find helpful.**
- 4. System registers the upvote.**
- 5. If a threshold is reached, the note is highlighted as a top note.**

Alternative Flows:

- **Duplicate upvote: If the user has already upvoted the note, the system prevents duplicate votes.**
- **Note is flagged: If a note is reported for misinformation, moderators may remove it.**

2.4 Display Highly Upvoted Community Notes

Use Case ID: #2-4

Use Case Name: Display Highly Upvoted Community Notes

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: System

Description: When a community note reaches a high number of upvotes, it is displayed as the primary fact-checking note for an article.

Preconditions:

- A community note has received enough upvotes to be marked as highly credible.

Postconditions:

- The note is prominently displayed under the article.
- Users see the most upvoted note first.

Priority: High

Frequency of Use: Medium

Flow of Events:

1. System monitors upvotes on community notes.
2. When a note reaches the required threshold, the system highlights it.
3. The note appears at the top of the article's community notes section.
4. Users see the most upvoted note first when checking the article.
5. System continues tracking votes and adjusts rankings if a new note surpasses the existing top note.

Alternative Flows:

- **Upvotes decrease:** If a note loses credibility (downvotes or reports), the system may remove it from the top position.
- **Multiple top notes:** If multiple notes reach the threshold, the system may rank them based on additional engagement factors.

3. User Management

3.1 Register an Account

Use Case ID: #3-1

Use Case Name: Register an Account

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: New User

Description: Users create a new account by providing necessary details.

Preconditions: User is not yet registered.

Postconditions: A new user account is created.

Priority: High

Frequency of Use: High

Flow of Events:

- 1. User navigates to the registration page.**
- 2. User enters details (username, email, password, etc.).**
- 3. System validates input and checks for duplicates.**
- 4. System sends verification email.**
- 5. User verifies email and completes registration.**

Alternative Flows:

- Invalid data: System prompts the user for corrections.**
- Email already in use: System notifies the user.**

3.2 Recover Lost Account

Use Case ID: #3-2

Use Case Name: Recover Lost Account

Created By: Riaz

Last Updated By: Riaz

Date Created: 07/03/2025

Date Last Updated: 07/03/2025

Actor: User, System

Description: Users can recover their account if they forget their credentials.

Preconditions: User has a registered email linked to the account.

Postconditions: User regains access to their account.

Priority: High

Frequency of Use: Medium

Flow of Events:

- 1. User navigates to the login page and clicks "Forgot Password."**
- 2. System prompts the user to enter their registered email.**
- 3. System sends an OTP to their registered email.**
- 4. Users are asked to login to their account using their OTP.**
- 5. Email prompts users to change their password after logging in using the OTP.**
- 6. Users successfully change their password.**

Alternative Flows:

- Invalid email: System notifies the user and prompts retry.**
- OTP expired: System allows the user to request a new OTP.**