

Faculdade de Engenharia da Universidade do Porto

**NewsNet**

**Product and Presentation**

Laboratório de Base de Dados e Aplicações Web

Turma 8 – Grupo 4

- Alexandre Henrique Silva dos Santos (up202108671@up.pt)

- João Pedro Cardoso de Couto ([up202006526@up.pt](mailto:up202006526@up.pt))

- Maria Carlota Gomes Ribeiro Matos Leite ([up202005428@up.pt](mailto:up202005428@up.pt))

- Maria Eduarda Pacheco Mendes Araújo ([up202004473@up.pt](mailto:up202004473@up.pt))

Porto, 22 de dezembro de 2024

**Index:**

**Index**

**A1: Project Presentation**

**A2: Actors and User Stories**

2.1. Actors

2.2. User Stories

2.2.1. User

2.2.2. Authenticated User

2.2.3. News Author

2.2.4. Comment Author

2.2.5. Unauthenticated User

2.2.6. System Manager

2.3. Supplementary Requirements

2.3.1. Business rules

2.3.2. Tecnical requirements

2.3.3. Restrictions

**A3: Information Architecture**

3.1. Sitemap

3.2. Wireframes

UI01. Homepage

UI10. Authors

UI11. Categories

UI12. News details

**A4: Conceptual Data Model**

4.1. Class diagram

4.2. Additional Business Rules

**A5: Relational Schema, validation and schema refinement**

5.1. Relational Schema

5.2. Domains

5.3. Schema validation

**A6: Indexes, triggers, transactions and database population**

6.1. Database Workload

6.2. Proposed Indices

6.2.1. Performance Indices

6.2.2. Full-text Search Indices

6.3. Triggers

6.4. Transactions

**A7: Web Resources Specification**

7.1. Overview

7.2. Permissions

**7.3. OpenAPI Specification (EM FALTA)**

**A8: Vertical Prototype**

8.1. Implemented Features

8.1.1. Implemented User Stories

8.2.1. Implemented Web Resources

8.2. Prototype

**A9: Product**

9.1. Installation

9.2. Usage

9.2.1. Administration Credentials

9.2.2. User Credentials

9.3. Application Help

9.4. Input Validation

9.5. Check Accessibility and and Usability

9.6. HTML & CSS Validation

9.7. Revisioons to the Project

9.8. Implementation Details

9.8.1. Libraries Used

9.8.2. User Stories

**A10: Presentation**

10.1. Product presentation

10.2. Video presentation

**Revision History**

**Annexes**

**A1: Project Presentation**

**NewsNet** is a web-based collaborative news management platform, developed by a group of students in the context of the LBAW course. It is designed for news lovers and aspiring journalists to have a place to create, share and engage in news.

This platform is designed to allow authenticated users to submit, vote, and comment on articles. Users can build social connections between them by following, and this can be used as a filter for news content based on personal networks. This system will provide robust news curation through a voting mechanism on news articles, as well as comments, with functionalities for system management and user interaction.

As an alternative route, if the user prefers to not create an account or log in, the system will be read-only, meaning that a selection of popular news articles (with bigger vote counts) will be visible, but the user cannot vote, comment or upload an article themselves.

This platform targets:

* news enthusiasts looking to stay on top of current events via community-curated content;
* writers and journalists who want to share their articles and have feedback;
* readers who want an active voice in news sharing, by voting, commenting, and sharing news;
* individuals who prefer a curated selection of news based on the contributions of their personal network.

**User profiles**

System managers:

* Oversee user activities, moderate content, and manage platform settings
* Insights on voting patterns, most active users, popular news articles and engagement trends

Common users:

* Submit news articles along with relevant media (text, images, videos)
* Vote on news articles and comments, enabling community-driven news ranking
* Comment on news articles and individual comments, enhancing discussion
* Establish connections with others to personalize their news feed based on their friends’ votes and contributions
* Filter news based on their friends’ activities, providing a more tailored news experience

**A2: Actors and User Stories**

**2.1. Actors**

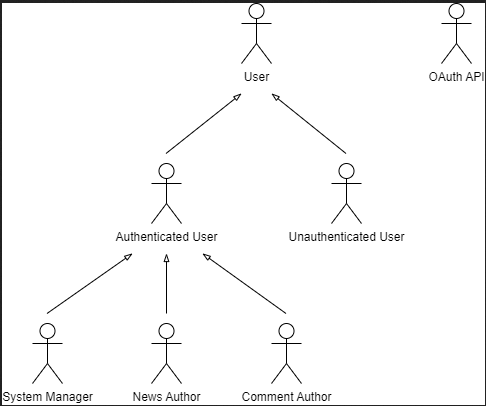


Figure 1: NewsNet actors

|  |  |
| --- | --- |
| Identifier | Description |
| User | Generic user that has access to public information, such as trending news. |
| Authenticated User | Authenticated user who can write articles, comment and vote on news, and follow other users and topics. |
| Unauthenticated User | Unauthenticated user that can register itself (sign-up) or sign-in in the system. |
| System Manager | Authenticated user that is responsible for the management of users and news topics. |
| News Author | Authenticated user who creates a news article and can then edit or delete it. |
| Comment Author | Authenticated user who comments on a news article and can then change or remove his action. |
| OAuthor API | External OAuth API that can be used to register or authenticate into the system. |

Table 1: NewsNet actors description

**2.2. User Stories**

**2.2.1. User**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US101 | View Top News Feed | High | As a User, I want to view a feed of top news, so I can quickly access the most relevant or popular articles. |
| US102 | View Recent News Feed | High | As a User, I want to view a feed with the latest news, so that I can keep up with the latest updates and news. |
| US103 | View News Item | High | As a User, I want to view the full content of a news item, so that I can read the story in detail. |
| US104 | View News Item Comments | High | As a User, I want to view other users comments on a news story, so that I can see their opinions and interact with the content more fully. |
| US105 | Search for News Items and Comments | High | As a User, I want to search for news and reviews, so I can find specific information or topics of interest. |
| US106 | Help | Medium | As a User, I want to have access to help messages, so I can clarify my doubts. |
| US107 | About Us | Medium | As a User, I want to have access to an About Us page, so I can learn more about the website. |
| US108 | Contacts | Medium | As a User, I want to be able to contact the administrators, so I can report a bug or make suggestions. |

Table 2: User’s User Stories

**2.2.2. Authenticated User**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US201 | View User News Feed | High | As an Authenticated User, I want to view a personalized news feed, so that I can see relevant news based on my interests, preferences, and past activities. |
| US202 | View News Item Details | High | As an Authenticated User, I want to view the full details of a news item, so that I can get more information about the author, publication date, votes, and other metadata. |
| US203 | View Comment Details | High | As an Authenticated User, I want to view the full details of a comment on a news item, so that I can know who made the comment, when it was posted, and how many votes it has received. |
| US204 | Create News Item | High | As an Authenticated User, I want to create and publish a news item, so that I can share relevant information or content with other users. |
| US205 | Vote on News Item | High | As an Authenticated User, I want to be able to vote on news items so that I can express my opinion about the content and influence its visibility. |
| US206 | Vote on Comment | High | As an Authenticated User, I want to be able to vote on comments on news items, so that I can express my opinion about the comment and influence its relevance. |
| US207 | Comment on News Item | High | As an Authenticated User, I want to comment on a news item, so that I can participate in discussions and share my opinions. |
| US208 | View Other Users' Profiles | High | As an Authenticated User, I want to view other users' profiles, so that I can see their activities, reputation, and published content. |
| US209 | View Reputation of Other Users | High | As an Authenticated User, I want to view the reputation of other users, so that I can assess their credibility and influence within the community. |
| US210 | View News Items Tags | High | As an Authenticated User, I want to view the tags associated with a news item, so that I can better understand the context and related topics of the content. |
| US211 | Follow/Unfollow Users | High | As an Authenticated User, I want to follow or unfollow other users, so that I can personalize my news feed and keep up with content from people I find interesting. |
| US212 | Follow/Unfollow Tags | High | As an Authenticated User, I want to follow or unfollow tags, so that I can personalize my news feed with topics of interest. |
| US213 | Follow Notification | High | As a Authenticated User, I want to be notified when someone follows me, so that I know if someone is interested in my content. |
| US214 | View My Profile | High | As an Authenticated User, I want to view my own profile, so that I can see my activity, reputation, and published content. |
| US215 | Edit My Profile | High | As an Authenticated User, I want to edit my own profile, so that I can change my information and profile picture. |
| US216 | Delete Account | High | As a Authenticated User, I want to be able delete my account, so that I can permanently remove my account from the system. |
| US217 | Logout | High | As an Authenticated User, I want to logout of my account, so that I can become un unauthenticated user. |
| US218 | Add/Remove News Item from Favorites | Medium | As an Authenticated User, I want to add or remove a News Item from my Favorites, so that I can save my favorite articles. |
| US219 | Report Users or Content | Medium | As an Authenticated User, I want to report users or content, so that I can help the System Managers maintain a disciplined environment. |
| US220 | Propose new Topic | Low | As an Authenticated User, I want to propose new news topics, so that I can better categorize a news article. |
| US221 | Appeal for Unblock | Low | As an Authenticated User, I want to appeal for unblock, so that I can be reconsidered if I'm unfairly blocked. |

Table 3: Authenticated User’s User Stories

**2.2.3. News Author**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US301 | Edit News Item | High | As a News Author, I want to be able to edit an existing news item so that I can correct information or update content according to new data or events. |
| US302 | Delete News Item | High | As a News Author, I want to delete a news item, so that I can remove information that is no longer relevant or has been published incorrectly. |
| US303 | Vote Notification | High | As a News Author, I want to be notified when someone votes on my article, so that I know that my content is being well received and engaged by others. |
| US304 | Comment Notification | High | As a News Author, I want to be notified when someone comments on my article, so I can follow the discussions and respond if necessary. |

Table 4: News Author’s User Stories

**2.2.4. Comment Author**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US401 | Edit Comment | High | As a Comment Author, I want to edit a comment I posted, so that I can correct errors or adjust the content according to new information or points of view.. |
| US402 | Delete Comment | High | As a Comment Author, I want to delete a comment I posted, so that I can remove content that I no longer consider relevant or appropriate. |
| US403 | Vote Notification | High | As a Comment Author, I want to be notified when someone votes on my comment, so that I know that my comment is being well received and engaged by others. |
| US404 | Comment Notification | High | As a Comment Author, I want to be notified when someone responds to my comment, so I can follow the discussions and respond if necessary. |

Table 5: Comment Author’s User Stories

**2.2.5. Unauthenticated User**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US501 | Sign-in | High | As a Unauthenticated User, I want to authenticate into the system, so that I can access privileged information. |
| US502 | Sign-up | High | As a Unauthenticated User, I want to register myself into the system, so that I can access privileged information. |
| US503 | Recover Password | High | As a Unauthenticated User, I want to be able to recover my account's password, so that I sign-in even if I have forgotten my password. |
| US504 | OAuth API Sign-in | Low | As a Unauthenticated User, I want to authenticate into the system using my Google account, so that I can easily access privileged information. |
| US505 | OAuth API Sign-up | Low | As a Unauthenticated User, I want to register a new account linked to my Google account, so that I do not need to create a whole new account to use the platform.. |

Table 6: Unauthenticated User’s User Stories

**2.2.6. System Manager**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US601 | Manage Administrator Accounts | High | As a System Manager, I want to manage administrator accounts, so that I can control access to the system and ensure only authorized personnel can perform administrative tasks. |
| US602 | Administer User Accounts | High | As a System Manager, I want to search, view, edit, and create user accounts, so that I can efficiently manage user information and permissions. |
| US603 | Block and Unblock User Accounts | High | As a System Manager, I want to block and unblock user accounts, so that I can temporarily restrict or restore access for users violating or adhering to platform policies. |
| US604 | Delete User Accounts | High | As a System Manager, I want to delete user accounts, so that I can permanently remove users from the platform if needed. |
| US605 | Manage Topic Proposals | High | As a System Manager, I want to manage topic proposals, so that I can analyze whether a topic should be added. |
| US606 | Delete Comments | Medium | As a System Manager, I want to delete comments, so that I can permanently remove comments deemed inappropriate. |
| US607 | Statistics | Medium | As a System Manager, I want to have access to the website statistics, so that I can get insights on voting patterns, most active users, popular news articles and engagement trends. |
| US608 | Manage Users or Content Reports | Low | As a System Manager, I want to manage users or content reports, so that I can analyze a report and take the appropriate repercussions. |
| US609 | Manage Unblock Appeals | Low | As a System Manager, I want to manage Unblock Appeals, so that I can analyze whether a user has been unfairly blocked. |

Table 7: System Manager’s User Stories

**2.3. Supplementary Requirements**

**2.3.1. Business rules**

|  |  |  |
| --- | --- | --- |
| Identifier | Name | Description |
| BR01 | Account Deletion | Upon account deletion, shared user data (e.g. comments, reviews, likes) is kept but is made anonymous. |
| BR02 | User Reputation | User reputation is dependent on the votes received on his posts or comments. |
| BR03 | Content Deletion | A post or comment cannot be deleted by its author if it has votes or comments. |
| BR04 | Content Edition | An edited post or comment is marked with a tag. |
| BR05 | Vote Own Content | A user is unable to vote on their own article or comment. |
| BR06 | News Topic | Every News Item must have an associated Topic. |
| BR07 | Date | Every comment date must be after the article date. |

Table 8: NewsNet Business Rules

**2.3.2. Technical requirements**

|  |  |  |
| --- | --- | --- |
| Identifier | Name | Description |
| TR01 | Performance | The system should have response times shorter than 2s to ensure the user's attention. |
| TR02 | Robustness | The system must be prepared to handle and continue operating when runtime errors occur. |
| TR03 | Scalability | The system must be prepared to deal with the growth in the number of users and their actions. |
| TR04 | Accessibility | The system must ensure that everyone can access the pages, regardless of whether they have any handicap or not, or the Web browser they use. |
| TR05 | Availability | The system must be available 99 percent of the time in each 24-hour period. |
| TR06 | Usability | The system should be simple and easy to use. The NewsNet system is designed to be used by media consumers from all ages, with or without technical experience, so a very good usability is a critical requirement. |
| TR07 | Web Application | The system should be implemented as a web application with dynamic pages (HTML5, JavaScript, CSS3 and PHP). It is critical that the NewsNet system is easily accessible from anywhere without the need to install specific applications or software, adopting standard web technologies. |
| TR08 | Portability | The server-side system should work across multiple platforms (Linux, Mac OS, etc.). The NewsNet system is destined for personal use. To make it easily available to a large user base, it should be platform-independent. |
| TR09 | Database | The PostgreSQL database management system must be used, with a version of 11 or higher. |
| TR10 | Security | The system shall protect information from unauthorized access through the use of an authentication and verification system |
| TR11 | Ethics | The system must respect the ethical principles in software development (for example, personal user details, or usage data, should not be collected nor shared without full acknowledgement and authorization from its owner). |

Tabla 9: NewsNet Technical Requirements

**2.3.3. Restrictions**

|  |  |  |
| --- | --- | --- |
| Identifier | Name | Description |
| C01 | Deadline | The system should have response times shorter than 2s to ensure the user's attention. |

Table 10: NewsNet Restrictions

**A3: Information Architecture**

**3.1. Sitemap**

A sitemap is a visual representation of the structure of a website or a system. It outlines all the pages, content, and sections within a website or application, showing how they’re interconnected.

Uma imagem com texto, captura de ecrã, diagrama, design

Descrição gerada automaticamente

Figure 2: NewsNet Sitemap

**3.2. Wireframes**

**UI01: Homepage**

Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamente

Figure 3: Homepage (UI01) wireframe

**UI10: Authors**

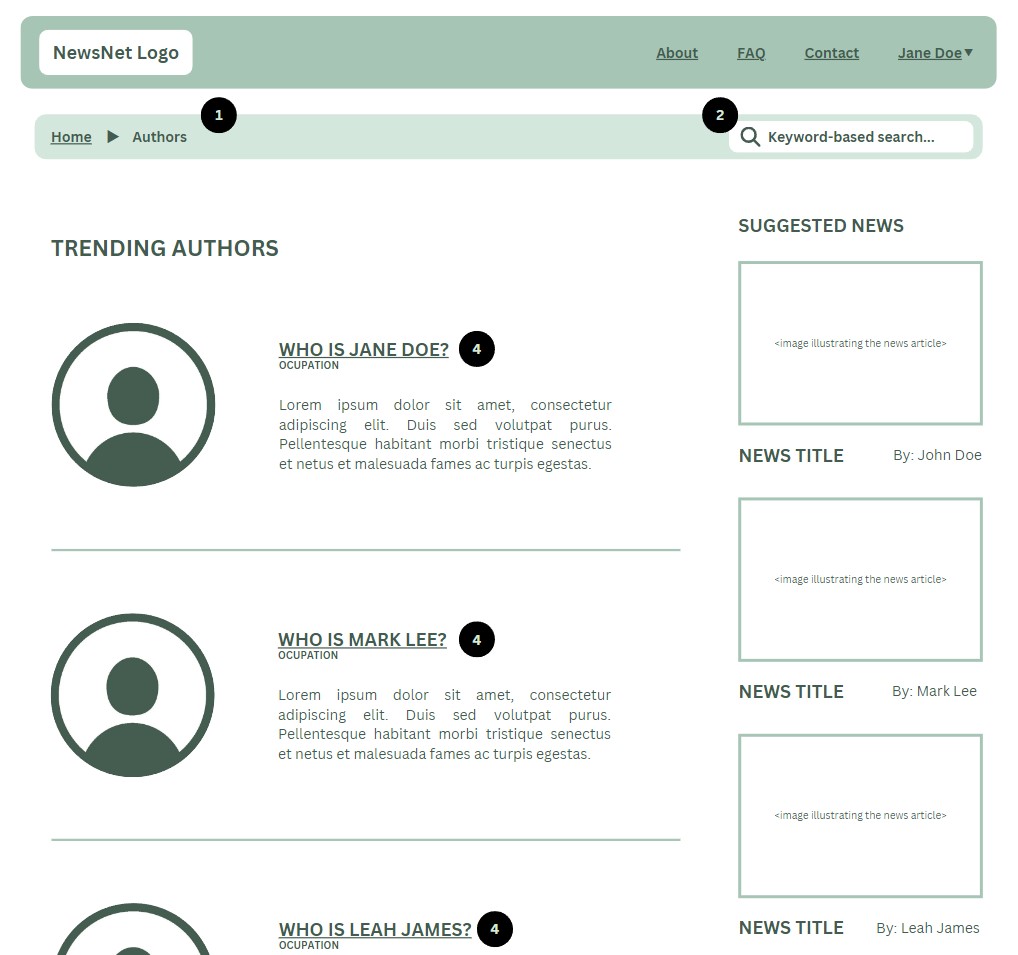
****

Figure 4: Authors (UI10) wireframe

**Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamenteUI11: Categories**

Figure 5: Categories (UI11) wireframe

**UI12: News details**

Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamente

Figure 6: News details (UI12) wireframe

**A4: Conceptual Data Model**

This section contains the description of the entities and relationships that exists to the NewsNet project and its databse specification.

**4.1. Class diagram**

This Figure shows a UML diagram of the primary organizational entities for the **NewsNet** platform, detailing their relationships, attributes and their domains, as well as the multiplicity of these relationships.

Uma imagem com texto, diagrama, Esquema, Desenho técnico

Descrição gerada automaticamente

Figura7: NewsNet Class Diagram

**4.2. Additional Business Rules**

Business rules or limitations that are unable to be depicted in the UML class diagram are included as UML notes in the diagram itself or as separate notes in this section.

|  |  |  |
| --- | --- | --- |
| Identifier | Name | Description |
| BR01 | Account Deletion | Upon account deletion, shared user data (e.g. comments, reviews, likes) is kept but is made anonymous. |
| BR02 | User Reputation | User reputation is dependent on the votes received on his posts or comments. |
| BR03 | Content Deletion | A post or comment cannot be deleted by its author if it has votes or comments. |
| BR04 | Vote Own Content | A user is unable to vote on their own article or comment. |
| BR05 | Comment Date | Every comment date must be after the article date. |
| BR06 | Self Report | A user is unable to report himself or his own content. |
| BR07 | Self Follow | A user is unable to follow himself. |
| BR08 | Notification Date | Every notification date must be the same as the date of the action being notified. |
| BR09 | Block and Unblock | Only System Managers can block or unblock users. |
| BR10 | Forced Deletion | System Managers can delete any user account or content. |
| BR11 | New Categories | Only System Managers can add new categories. |
| BR12 | Edition Date | If it exists, every edition date must be after the creation date. |
| BR13 | Content Edition | Only the owner of a content can edit it or, apart from System Managers, delete it. |
| BR14 | Single Vote | A user can only vote once on a post or comment. |
| BR15 | Automatic Notifications | Notifications should be added automatically when the action to be notified is processed. |
| BR16 | Reply Date | Every reply date must be after the parent comment date. |

Table 11: NewsNet Business Rules

**A5: Relational Schema, validation and schema refinement**

**5.1. Relational Schema**

|  |  |
| --- | --- |
| Domain reference | Relation Compact Notation |
| R01 | users(user\_id, username **UK NN**, email **UK NN**, password **NN**, reputation **DF** 0, created\_time **DF** today) |
| R02 | system\_managers(sm\_id → users) |
| R03 | follows(follower\_id → users, followed\_id → users, **CK** follower\_id <> followed\_id) |
| R04 | categories(category\_id, name **UK NN**) |
| R05 | user\_category(user\_id → users, category\_id → categories) |
| R06 | posts(post\_id, user\_id → users, title **NN**, body **NN**, created\_time **DF** today, updated\_time, **CK** (updated\_time IS NULL OR updated\_time > created\_time)) |
| R07 | post\_categories(post\_id → posts, category\_id → categories) |
| R08 | comments(comment\_id, post\_id → posts, user\_id → users, body **NN**, created\_time **DF** today, updated\_time, **CK** (updated\_time IS NULL OR updated\_time > created\_time)) |
| R09 | replies(parent\_comment\_id → comments, comment\_id → comments, **CK** parent\_comment\_id <> comment\_id) |
| R10 | post\_votes(vote\_id, user\_id → users, post\_id → posts, is\_like **NN**, time **DF** today) |
| R11 | comment\_votes(vote\_id, user\_id → users, comment\_id → comments, is\_like **NN**, time **DF** today) |
| R12 | user\_favorites(user\_id → users, post\_id → posts) |
| R13 | follow\_notification(notification\_id, user\_id → users, follower\_id → users, viewed **DF** FALSE, time **DF** today) |
| R14 | vote\_notification(notification\_id, user\_id → users, post\_id → posts, comment\_id → comments, viewed **DF** FALSE, time **DF** today) |
| R15 | comment\_notification(notification\_id, user\_id → users, post\_id → posts, parent\_comment\_id → comments, viewed **DF** FALSE, time **DF** today) |
| R16 | post\_notification(notification\_id, user\_id → users, author\_id → users, post\_id → posts, viewed **DF** FALSE, time **DF** today) |
| R17 | user\_report(report\_id, reporter\_id → users, reported\_id → users, reason **NN**, time **DF** today, **CK** reporter\_id <> reported\_id) |
| R18 | post\_report(report\_id, reporter\_id → users, reported\_id → users, post\_id → posts, reason **NN**, time **DF** today) |
| R19 | comment\_report(report\_id, reporter\_id → users, reported\_id → users, comment\_id → comments, reason **NN**, time **DF** today) |
| R20 | blocked\_user(blocked\_id → users, blocked\_at **DF** today, reason, report\_id → user\_report) |

Table 12: NewsNet Relational Schema

**5.2. Domains**

|  |  |
| --- | --- |
| Domain Name | Domain Specification |
| Today | DATE DEFAULT CURRENT\_DATE |

Table 13: NewsNet Domains

**5.3. Schema validation**

|  |  |
| --- | --- |
| TABLE R01 | users |
| Keys | { user\_id }, { username }, { email } |
| Functional Dependencies: |  |
| FD0101 | {user\_id} → {username, email, password, reputation, created\_time} |
| FD0102 | {username} → {user\_id, email, password, reputation, created\_time} |
| FD0103 | {email} → {user\_id, username, password, reputation, created\_time} |
| NORMAL FORM | BCNF |

Table 14: NewsNet User Schema Validation

|  |  |
| --- | --- |
| TABLE R02 | system\_managers |
| Keys | { sm\_id } |
| Functional Dependencies: | None |
| NORMAL FORM | BCNF |

Table 15: NewsNet Sytem Manager Schema Validation

|  |  |
| --- | --- |
| TABLE R03 | follows |
| Keys | { follower\_id }, { followed\_id } |
| Functional Dependencies: | None |
| NORMAL FORM | BCNF |

Table 16: NewsNet Follow Schema Validation

|  |  |
| --- | --- |
| TABLE R04 | categories |
| Keys | { category\_id } |
| Functional Dependencies: |  |
| FD0401 | {category\_id} → {name} |
| NORMAL FORM | BCNF |

Table 17: NewsNet Category Schema Validation

|  |  |
| --- | --- |
| TABLE R05 | user\_category |
| Keys | { user\_id }, { category\_id } |
| Functional Dependencies: | None |
| NORMAL FORM | BCNF |

Table 18: NewsNet User Category Schema Validation

|  |  |
| --- | --- |
| TABLE R06 | posts |
| Keys | { post\_id } |
| Functional Dependencies: |  |
| FD0601 | {post\_id} → {user\_id, title, body, created\_time, updated\_time} |
| NORMAL FORM | BCNF |

Table 19: NewsNet Post Schema Validation

|  |  |
| --- | --- |
| TABLE R07 | post\_categories |
| Keys | { post\_id }, { category\_id } |
| Functional Dependencies: | None |
| NORMAL FORM | BCNF |

Table 20: NewsNet Post Category Schema Validation

|  |  |
| --- | --- |
| TABLE R08 | comments |
| Keys | { comment\_id } |
| Functional Dependencies: |  |
| FD0801 | {comment\_id} → {post\_id, user\_id, body, created\_time, updated\_time} |
| NORMAL FORM | BCNF |

Table 21: NewsNet Comment Schema Validation

|  |  |
| --- | --- |
| TABLE R09 | replies |
| Keys | { parent\_comment\_id }, { comment\_id } |
| Functional Dependencies: | None |
| NORMAL FORM | BCNF |

Table 22: NewsNet Reply Schema Validation

|  |  |
| --- | --- |
| TABLE R10 | post\_votes |
| Keys | { vote\_id } |
| Functional Dependencies: |  |
| FD1001 | {vote\_id} → {user\_id, post\_id, is\_like, time} |
| NORMAL FORM | BCNF |

Table 23: NewsNet Post Vote Schema Validation

|  |  |
| --- | --- |
| TABLE R11 | comment\_votes |
| Keys | { vote\_id } |
| Functional Dependencies: |  |
| FD1001 | {vote\_id} → {user\_id, comment\_id, is\_like, time} |
| NORMAL FORM | BCNF |

Table 24: NewsNet Comment Vote Schema Validation

|  |  |
| --- | --- |
| TABLE R12 | user\_favorites |
| Keys | { user\_id }, { post\_id } |
| Functional Dependencies: | None |
| NORMAL FORM | BCNF |

Table 25: NewsNet User Favorite Schema Validation

|  |  |
| --- | --- |
| TABLE R13 | follow\_notification |
| Keys | { notification\_id } |
| Functional Dependencies: |  |
| FD1001 | {notification\_id} → {user\_id, follower\_id, viewed, time} |
| NORMAL FORM | BCNF |

Table 26: NewsNet Follow Notification Schema Validation

|  |  |
| --- | --- |
| TABLE R14 | vote\_notification |
| Keys | { notification\_id } |
| Functional Dependencies: |  |
| FD1001 | {notification\_id} → {user\_id, post\_id, comment\_id, viewed, time} |
| NORMAL FORM | BCNF |

Table 27: NewsNet Vote Notification Schema Validation

|  |  |
| --- | --- |
| TABLE R15 | comment\_notification |
| Keys | { notification\_id } |
| Functional Dependencies: |  |
| FD1001 | {notification\_id} → {user\_id, post\_id, parent\_comment\_id, viewed, time} |
| NORMAL FORM | BCNF |

Table 28: NewsNet Comment Notification Schema Validation

|  |  |
| --- | --- |
| TABLE R16 | post\_notification |
| Keys | { notification\_id } |
| Functional Dependencies: |  |
| FD1001 | {notification\_id} → {user\_id, author\_id, post\_id, viewed, time} |
| NORMAL FORM | BCNF |

Table 29: NewsNet Post Notification Schema Validation

|  |  |
| --- | --- |
| TABLE R17 | user\_report |
| Keys | { report\_id } |
| Functional Dependencies: |  |
| FD1001 | {report\_id} → {reporter\_id, reported\_id, reason, time} |
| NORMAL FORM | BCNF |

Table 30: NewsNet User Report Schema Validation

|  |  |
| --- | --- |
| TABLE R18 | post\_report |
| Keys | { report\_id } |
| Functional Dependencies: |  |
| FD1001 | {report\_id} → {reporter\_id, reported\_id, post\_id, reason, time} |
| NORMAL FORM | BCNF |

Table 31: NewsNet Post Report Schema Validation

|  |  |
| --- | --- |
| TABLE R19 | comment\_report |
| Keys | { report\_id } |
| Functional Dependencies: |  |
| FD1001 | {report\_id} → {reporter\_id, reported\_id, comment\_id, reason, time} |
| NORMAL FORM | BCNF |

Table 32: NewsNet Comment Report Schema Validation

|  |  |
| --- | --- |
| TABLE R20 | blocked\_user |
| Keys | { blocked\_id } |
| Functional Dependencies: |  |
| FD1001 | {blocked\_id} → {blocked\_at, reason, report\_id} |
| NORMAL FORM | BCNF |

Table 33: NewsNet Blocked User Schema Validation

**A6: Indexes, triggers, transactions and database population**

**6.1. Database Workload**

|  |  |  |  |
| --- | --- | --- | --- |
| Relation reference | Relation Name | Order of magnitude | Estimated growth |
| R01 | users | 10k | 10 / day |
| R02 | system\_managers | 100 | 1 / month |
| R03 | follows | 100k | 1k / day |
| R04 | categories | 100 | 1 / month |
| R05 | user\_category | 10k | 100 / day |
| R06 | posts | 100k | 10k / day |
| R07 | post\_categories | 10k | 1k / day |
| R08 | comments | 100k | 10k / day |
| R09 | replies | 10k | 1k / day |
| R10 | post\_votes | 100k | 1k / day |
| R11 | comment\_votes | 100k | 1k / day |
| R12 | user\_favorites | 10k | 100 / day |
| R13 | follow\_notification | 100k | 1k / day |
| R14 | vote\_notification | 100k | 1k / day |
| R15 | comment\_notification | 100k | 1k / day |
| R16 | post\_notification | 10k | 100 / day |
| R17 | user\_report | 1k | 1 / day |
| R18 | post\_report | 10k | 1 / day |
| R19 | comment\_report | 10k | 10 / day |
| R20 | blocked\_users | 100 | 1 / day |

Table 34: NewsNet Workload

**6.2. Performance Indices**

**6.2.1. Performance Indices**

|  |  |
| --- | --- |
| Index | IDX01 |
| Relation | posts |
| Attribute | user\_id |
| Type | Hash |
| Cardinality | medium |
| Clustering | No |
| Justification | On a Collaborative News platform, personalized feeds and news search by author should be important, which is why this index enhances the performance of queries that fetch all posts created by a specific user. |
| SQL code | CREATE INDEX index\_posts\_user\_id ON posts USING hash(user\_id); |

Table 35: NewsNet Posts User Index

|  |  |
| --- | --- |
| Index | IDX02 |
| Relation | comments |
| Attribute | user\_id |
| Type | Hash |
| Cardinality | high |
| Clustering | No |
| Justification | As a high number of comments on the platform is expected, this index optimizes the retrieval of comments for a specific post. |
| SQL code | CREATE INDEX index\_comments\_post\_id ON comments USING hash(post\_id); |

Table 36: NewsNet Posts Comments Index

|  |  |
| --- | --- |
| Index | IDX03 |
| Relation | posts |
| Attribute | create\_time |
| Type | B-tree |
| Cardinality | high |
| Clustering | Yes |
| Justification | In order to retrieve recent posts efficiently, this index proves to be crucial for speeding up queries that order posts by creation time. Clustering ensures that similar timestamps are kept together, optimizing access time for recent content. |
| SQL code | CREATE INDEX index\_posts\_created\_time ON posts USING btree(created\_time);  CLUSTER posts USING index\_posts\_created\_time; |

Table 37: NewsNet Post Created Time Index

**6.2.2. Full-text Search Indices**

|  |  |
| --- | --- |
| Index | IDX04 |
| Relation | posts |
| Attribute | title, body |
| Type | GIN |
| Clustering | No |
| Justification | To provide full-text search features to look for posts based on matching titles or body content. The index type is GIN because the indexed fields are not expected to change often. |
| SQL code | ALTER TABLE posts  ADD COLUMN tsvectors TSVECTOR;  CREATE FUNCTION posts\_search\_update()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF TG\_OP = 'INSERT' THEN  NEW.tsvectors = (  setweight(to\_tsvector('english', NEW.title), 'A') ||  setweight(to\_tsvector('english', NEW.body), 'B')  );  END IF;  IF TG\_OP = 'UPDATE' THEN  IF (NEW.title <> OLD.title OR NEW.body <> OLD.body) THEN  NEW.tsvectors = (  setweight(to\_tsvector('english', NEW.title), 'A') ||  setweight(to\_tsvector('english', NEW.body), 'B')  );  END IF;  END IF;  RETURN NEW;  END  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER posts\_search\_update  BEFORE INSERT OR UPDATE ON posts  FOR EACH ROW  EXECUTE PROCEDURE posts\_search\_update();  CREATE INDEX search\_idx ON posts USING GIN (tsvectors); |

Table 38: NewsNet Full-text Search Index

**6.2.3. Triggers**

|  |  |
| --- | --- |
| Trigger | TRIGGER01 |
| Description | A post cannot be deleted by its author if it has votes or comments (BR03). |
| SQL code | CREATE FUNCTION prevent\_post\_deletion()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF (SELECT COUNT(\*) FROM comments WHERE post\_id = OLD.post\_id) > 0 OR  (SELECT COUNT(\*) FROM post\_votes WHERE post\_id = OLD.post\_id) > 0  THEN  RAISE EXCEPTION 'Cannot delete content with existing votes or comments.';  END IF;  RETURN OLD;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER prevent\_post\_deletion  BEFORE DELETE ON posts  FOR EACH ROW  EXECUTE PROCEDURE prevent\_post\_deletion(); |

Table 39: NewsNet Prevent Post Deletion Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER02 |
| Description | A comment cannot be deleted by its author if it has votes or replies (BR03). |
| SQL code | CREATE FUNCTION prevent\_comment\_deletion()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF (SELECT COUNT(\*) FROM replies WHERE parent\_comment\_id = OLD.comment\_id) > 0 OR  (SELECT COUNT(\*) FROM comment\_votes WHERE comment\_id = OLD.comment\_id) > 0  THEN  RAISE EXCEPTION 'Cannot delete content with existing votes or comments.';  END IF;  RETURN OLD;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER prevent\_comment\_deletion  BEFORE DELETE ON comments  FOR EACH ROW  EXECUTE PROCEDURE prevent\_comment\_deletion(); |

Table 40: NewsNet Prevent Comment Deletion Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER03 |
| Description | A Notification should be added automatically when a user is followed (BR15). |
| SQL code | CREATE FUNCTION create\_follow\_notification()  RETURNS TRIGGER AS  $BODY$  BEGIN  INSERT INTO follow\_notification (user\_id, follower\_id, time)  VALUES (NEW.followed\_id, NEW.follower\_id, NOW());  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER trigger\_follow\_notification  AFTER INSERT ON follows  FOR EACH ROW  EXECUTE PROCEDURE create\_follow\_notification(); |

Table 41: NewsNet Follow Notification Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER04 |
| Description | A Notification should be added automatically when a user's post is voted up (BR15). |
| SQL code | CREATE FUNCTION create\_vote\_notification\_post()  RETURNS TRIGGER AS  $BODY$  BEGIN  INSERT INTO vote\_notification (user\_id, post\_id, time)  VALUES ((SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id), NEW.post\_id, NOW());  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER trigger\_vote\_notification\_post  AFTER INSERT ON post\_votes  FOR EACH ROW  EXECUTE PROCEDURE create\_vote\_notification\_post(); |

Table 42: NewsNet Post Vote Notification Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER05 |
| Description | A Notification should be added automatically when a user's comment is voted up (BR15). |
| SQL code | CREATE FUNCTION create\_vote\_notification\_comment()  RETURNS TRIGGER AS  $BODY$  BEGIN  INSERT INTO vote\_notification (user\_id, comment\_id, time)  VALUES ((SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id), NEW.comment\_id, NOW());  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER trigger\_vote\_notification\_comment  AFTER INSERT ON comment\_votes  FOR EACH ROW  EXECUTE PROCEDURE create\_vote\_notification\_comment(); |

Table 43: NewsNet Comment Vote Notification Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER06 |
| Description | A Notification should be added automatically when a user's post is commented on (BR15). |
| SQL code | CREATE FUNCTION create\_comment\_notification()  RETURNS TRIGGER AS  $BODY$  BEGIN  INSERT INTO comment\_notification (user\_id, post\_id, comment\_id, time)  VALUES ((SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id), NEW.post\_id, NEW.comment\_id, NOW());  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER trigger\_comment\_notification  AFTER INSERT ON comments  FOR EACH ROW  EXECUTE PROCEDURE create\_comment\_notification(); |

Table 44: NewsNet Comment Notification Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER07 |
| Description | A Notification should be added automatically when a user's comment is replied (BR15). |
| SQL code | CREATE FUNCTION create\_reply\_notification()  RETURNS TRIGGER AS  $BODY$  BEGIN  INSERT INTO comment\_notification (user\_id, post\_id, parent\_comment\_id, comment\_id, time)  VALUES ((SELECT user\_id FROM comments WHERE comment\_id = NEW.parent\_comment\_id), (SELECT post\_id FROM comments WHERE comment\_id = NEW.comment\_id), NEW.parent\_comment\_id, NEW.comment\_id, NOW());  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER trigger\_reply\_notification  AFTER INSERT ON replies  FOR EACH ROW  EXECUTE PROCEDURE create\_reply\_notification(); |

Table 45: NewsNet Reply Notification Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER08 |
| Description | A Notification should be added automatically when a followed user posts (BR15). |
| SQL code | CREATE FUNCTION create\_post\_notification()  RETURNS TRIGGER AS  $BODY$  BEGIN  INSERT INTO post\_notification (user\_id, author\_id, post\_id, time)  SELECT follower\_id, NEW.user\_id, NEW.post\_id, NOW()  FROM follows  WHERE followed\_id = NEW.user\_id;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER trigger\_post\_notification  AFTER INSERT ON posts  FOR EACH ROW  EXECUTE PROCEDURE create\_post\_notification(); |

Table 46: NewsNet Post Notification Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER09 |
| Description | User reputation is dependent on the votes received on his posts (BR02). |
| SQL code | CREATE FUNCTION update\_user\_reputation()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF TG\_TABLE\_NAME = 'post\_votes' THEN  IF NEW.is\_like THEN  UPDATE users  SET reputation = reputation + 1  WHERE user\_id = (SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id);  ELSE  UPDATE users  SET reputation = reputation - 1  WHERE user\_id = (SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id);  END IF;  ELSIF TG\_TABLE\_NAME = 'comment\_votes' THEN  IF NEW.is\_like THEN  UPDATE users  SET reputation = reputation + 1  WHERE user\_id = (SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id);  ELSE  UPDATE users  SET reputation = reputation - 1  WHERE user\_id = (SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id);  END IF;  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  -- Trigger for post votes  CREATE TRIGGER update\_reputation\_post\_vote  AFTER INSERT ON post\_votes  FOR EACH ROW  EXECUTE PROCEDURE update\_user\_reputation(); |

Table 47: NewsNet User Reputation (posts) Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER10 |
| Description | User reputation is dependent on the votes received on his comments (BR02) |
| SQL code | CREATE FUNCTION update\_user\_reputation()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF TG\_TABLE\_NAME = 'post\_votes' THEN  IF NEW.is\_like THEN  UPDATE users  SET reputation = reputation + 1  WHERE user\_id = (SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id);  ELSE  UPDATE users  SET reputation = reputation - 1  WHERE user\_id = (SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id);  END IF;  ELSIF TG\_TABLE\_NAME = 'comment\_votes' THEN  IF NEW.is\_like THEN  UPDATE users  SET reputation = reputation + 1  WHERE user\_id = (SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id);  ELSE  UPDATE users  SET reputation = reputation - 1  WHERE user\_id = (SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id);  END IF;  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  -- Trigger for comment votes  CREATE TRIGGER update\_reputation\_comment\_vote  AFTER INSERT ON comment\_votes  FOR EACH ROW  EXECUTE PROCEDURE update\_user\_reputation(); |

Table 48: NewsNet User Reputation (comments) Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER11 |
| Description | Every comment date must be after the article date (BR05). |
| SQL code | CREATE FUNCTION check\_comment\_date()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF NEW.created\_time < (SELECT created\_time FROM posts WHERE post\_id = NEW.post\_id) THEN  RAISE EXCEPTION 'Comment date must be after the article date';  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER check\_comment\_date\_trigger  BEFORE INSERT OR UPDATE ON comments  FOR EACH ROW EXECUTE PROCEDURE check\_comment\_date(); |

Table 49: NewsNet Comment Date Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER012 |
| Description | Every reply date must be after the parent comment date (BR16). |
| SQL code | CREATE FUNCTION check\_reply\_date()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF (SELECT created\_time FROM comments WHERE comment\_id = NEW.comment\_id) < (SELECT created\_time FROM comments WHERE comment\_id = NEW.parent\_comment\_id) THEN  RAISE EXCEPTION 'Reply date must be after the parent comment date';  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER check\_reply\_date\_trigger  BEFORE INSERT OR UPDATE ON replies  FOR EACH ROW EXECUTE PROCEDURE check\_reply\_date(); |

Table 50: NewsNet Reply Date Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER13 |
| Description | A user is unable to vote on their own article (BR04). |
| SQL code | CREATE FUNCTION no\_self\_vote\_post()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF NEW.user\_id = (SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id) THEN  RAISE EXCEPTION 'User cannot vote on their own post';  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER no\_self\_vote\_post\_trigger  BEFORE INSERT OR UPDATE ON post\_votes  FOR EACH ROW EXECUTE PROCEDURE no\_self\_vote\_post(); |

Table 51: NewsNet No Post Self Vote Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER14 |
| Description | A user is unable to vote on their own comment (BR04). |
| SQL code | CREATE FUNCTION no\_self\_vote\_comment()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF NEW.user\_id = (SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id) THEN  RAISE EXCEPTION 'User cannot vote on their own comment';  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER no\_self\_vote\_comment\_trigger  BEFORE INSERT OR UPDATE ON comment\_votes  FOR EACH ROW EXECUTE PROCEDURE no\_self\_vote\_comment(); |

Table 52: NewsNet No Comment Self Vote Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER15 |
| Description | A user is unable to report his own post (BR06). |
| SQL code | CREATE FUNCTION no\_self\_report\_post()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF NEW.reporter\_id = (SELECT user\_id FROM posts WHERE post\_id = NEW.post\_id) THEN  RAISE EXCEPTION 'User cannot report their own post';  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER no\_self\_report\_post\_trigger  BEFORE INSERT OR UPDATE ON post\_report  FOR EACH ROW EXECUTE PROCEDURE no\_self\_report\_post(); |

Table 53: NewsNet No Post Self Report Trigger

|  |  |
| --- | --- |
| Trigger | TRIGGER16 |
| Description | A user is unable to report his own comment (BR06). |
| SQL code | CREATE FUNCTION no\_self\_report\_comment()  RETURNS TRIGGER AS  $BODY$  BEGIN  IF NEW.reporter\_id = (SELECT user\_id FROM comments WHERE comment\_id = NEW.comment\_id) THEN  RAISE EXCEPTION 'User cannot report their own comment';  END IF;  RETURN NEW;  END;  $BODY$  LANGUAGE plpgsql;  CREATE TRIGGER no\_self\_report\_comment\_trigger  BEFORE INSERT OR UPDATE ON comment\_report  FOR EACH ROW EXECUTE PROCEDURE no\_self\_report\_comment(); |

Table 54: NewsNet No Comment Self Report Trigger

**6.2.3. Transactions**

|  |  |
| --- | --- |
| Transaction | TRAN01 |
| Description | Insert a new post by a user. |
| Justification | Ensures that all steps in creating a post, including inserting the post and associating categories, are committed as a unit. REPEATABLE READ is chosen to prevent inconsistent reads if other posts are being created concurrently. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  INSERT INTO posts (user\_id, title, body, created\_time)  VALUES ($user\_id, $title, $body, NOW());  INSERT INTO post\_categories (post\_id, category\_id)  VALUES (currval('posts\_post\_id\_seq'), $category\_id);  END TRANSACTION; |

Table 55: NewsNet Create Post Transaction

|  |  |
| --- | --- |
| Transaction | TRAN02 |
| Description | Retrieve follow notifications for a user in a read-only transaction. |
| Justification | To provide consistent data while ensuring high performance, a read-only transaction with SERIALIZABLE READ ONLY isolation level is used to prevent updates during the transaction. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  SELECT \* FROM follow\_notification  WHERE user\_id = $user\_id  ORDER BY created\_time DESC;  END TRANSACTION; |

Table 56: NewsNet Get Follow Notification Transaction

|  |  |
| --- | --- |
| Transaction | TRAN03 |
| Description | Retrieve vote notifications for a user in a read-only transaction. |
| Justification | To provide consistent data while ensuring high performance, a read-only transaction with SERIALIZABLE READ ONLY isolation level is used to prevent updates during the transaction. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  SELECT \* FROM vote\_notification  WHERE user\_id = $user\_id  ORDER BY created\_time DESC;  END TRANSACTION; |

Table 57: NewsNet Get Vote Notification Transaction

|  |  |
| --- | --- |
| Transaction | TRAN04 |
| Description | Retrieve comment notifications for a user in a read-only transaction. |
| Justification | To provide consistent data while ensuring high performance, a read-only transaction with SERIALIZABLE READ ONLY isolation level is used to prevent updates during the transaction. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  SELECT \* FROM comment\_notification  WHERE user\_id = $user\_id  ORDER BY created\_time DESC;  END TRANSACTION; |

Table 58: NewsNet Get Comment Notification Transaction

|  |  |
| --- | --- |
| Transaction | TRAN05 |
| Description | Retrieve post notifications for a user in a read-only transaction. |
| Justification | To provide consistent data while ensuring high performance, a read-only transaction with SERIALIZABLE READ ONLY isolation level is used to prevent updates during the transaction. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  SELECT \* FROM post\_notification  WHERE user\_id = $user\_id  ORDER BY created\_time DESC;  END TRANSACTION; |

Table 59: NewsNet Get Post Notification Transaction

|  |  |
| --- | --- |
| Transaction | TRAN06 |
| Description | Retrieve all comments for a specific post in a read-only transaction. |
| Justification | Since this is a read-only operation with no risk of modification, SERIALIZABLE READ ONLY isolation level is used to ensure consistent reads. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  SELECT \*  FROM comments  WHERE post\_id = $post\_id  ORDER BY created\_time ASC;  END TRANSACTION; |

Table 60: NewsNet Get Post Comments Transaction

|  |  |
| --- | --- |
| Transaction | TRAN07 |
| Description | Insert a reply to an existing comment on a post. |
| Justification | Ensures that the reply is inserted in a consistent manner, maintaining a strict order. REPEATABLE READ ensures no concurrent replies cause inconsistent replies within the transaction. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  INSERT INTO comments (post\_id, user\_id, body, created\_time)  VALUES ($post\_id, $user\_id, $body, NOW());  INSERT INTO replies (parent\_comment\_id, comment\_id)  VALUES ($parent\_comment\_id, currval('comments\_comment\_id\_seq'));  END TRANSACTION; |

Table 61: NewsNet Create Comment Reply Transaction

|  |  |
| --- | --- |
| Transaction | TRAN08 |
| Description | Retrieve all comment replies for a specific comment in a read-only transaction. |
| Justification | Since this is a read-only operation with no risk of modification, SERIALIZABLE READ ONLY isolation level is used to ensure consistent reads. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  SELECT \*  FROM replies  WHERE parent\_comment\_id = $parent\_comment\_id  ORDER BY created\_time ASC;  END TRANSACTION; |

Table 62: NewsNet Get Comment Replies Transaction

|  |  |
| --- | --- |
| Transaction | TRAN09 |
| Description | Insert or update a user's vote on a post. |
| Justification | Ensures consistent voting records by preventing race conditions where multiple votes from the same user could occur simultaneously. REPEATABLE READ is used to prevent inconsistent voting behavior. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  INSERT INTO post\_votes (user\_id, post\_id, is\_like, time)  VALUES ($user\_id, $post\_id, $is\_like, NOW())  ON CONFLICT (user\_id, post\_id)  DO UPDATE SET is\_like = EXCLUDED.is\_like, time = NOW();  END TRANSACTION; |

Table 63: NewsNet Create Post Vote Transaction

|  |  |
| --- | --- |
| Transaction | TRAN10 |
| Description | Mark a follow notification as viewed for a user. |
| Justification | This transaction ensures that marking a follow notification as viewed happens atomically. REPEATABLE READ is chosen to prevent inconsistent state if multiple notifications are marked read simultaneously. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  UPDATE follow\_notification  SET viewed = TRUE  WHERE notification\_id = $notification\_id AND user\_id = $user\_id;  END TRANSACTION; |

Table 64: NewsNet Mark Follow Notification As Viewed Transaction

|  |  |
| --- | --- |
| Transaction | TRAN11 |
| Description | Mark a vote notification as viewed for a user. |
| Justification | This transaction ensures that marking a vote notification as viewed happens atomically. REPEATABLE READ is chosen to prevent inconsistent state if multiple notifications are marked read simultaneously. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  UPDATE vote\_notification  SET viewed = TRUE  WHERE notification\_id = $notification\_id AND user\_id = $user\_id;  END TRANSACTION; |

Table 65: NewsNet Mark Vote Notification As Viewed Transaction

|  |  |
| --- | --- |
| Transaction | TRAN12 |
| Description | Mark a comment notification as viewed for a user. |
| Justification | This transaction ensures that marking a comment notification as viewed happens atomically. REPEATABLE READ is chosen to prevent inconsistent state if multiple notifications are marked read simultaneously. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  UPDATE comment\_notification  SET viewed = TRUE  WHERE notification\_id = $notification\_id AND user\_id = $user\_id;  END TRANSACTION; |

Table 66: NewsNet Mark Comment Notification As Viewed Transaction

|  |  |
| --- | --- |
| Transaction | TRAN13 |
| Description | Mark a post notification as viewed for a user. |
| Justification | This transaction ensures that marking a post notification as viewed happens atomically. REPEATABLE READ is chosen to prevent inconsistent state if multiple notifications are marked read simultaneously. |
| Isolation Level | REPEATABLE READ |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  UPDATE post\_notification  SET viewed = TRUE  WHERE notification\_id = $notification\_id AND user\_id = $user\_id;  END TRANSACTION; |

Table 67: NewsNet Mark Post Notification As Viewed Transaction

|  |  |
| --- | --- |
| Transaction | TRAN14 |
| Description | Delete a user from the database. |
| Justification | To maintain the integrity of the database, it is crucial to use a transaction when deleting a user. If the transaction fails, a ROLLBACK will be executed to revert all changes, thus maintaining data consistency. |
| Isolation Level | SERIALIZABLE READ ONLY |
| SQL code | BEGIN TRANSACTION;  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY;  DELETE FROM users WHERE user\_id = $user\_id;  END TRANSACTION; |

Table 68: NewsNet Delete User Transaction

**A7: Web Resources Specification**

**7.1. Overview**

|  |  |
| --- | --- |
| Module | Description |
| M01: Authentication | Web resources associated with user authentication. Includes the following system features: user registration, login, logout and credential recovery. |
| M02: Users | Web resources associated with user profile management and interactions. Includes the following system features: view, delete and edit user profile, follow and unfollow users, follow and unfollow categories, see notifications and report users. |
| M03: News Posts | Web resources associated with creating, managing and interacting with news posts. Includes the following system features: create, view, edit and delete posts, vote on posts, manage favorite posts, categorize posts and report posts. |
| M04: Comments | Web resources associated with comments on posts. Includes the following system features: add, edit and delete comments, reply to comments, vote on comments and report comments. |
| M05: Search | Web resources associated with searching various content in the platform. Includes the following system features: search posts, comments, users and categories by keywords. |
| M06: System Management | Web resources associated with user and content management for system managers. Includes the following system features: view and manage users (block/unblock), view and resolve reports and manage categories (add/delete). |
| M07: Notifications | Web resources associated with user notifications. Includes the following system features: delete and edit (viewed/not viewed) follow, vote, comment and post notifications. |
| M08: Static Pages | Web resources associated with static pages. Includes the following system features: view about page. |

Table 69: NewsNet Modules

**7.2. Permissions**

|  |  |  |
| --- | --- | --- |
| Identifier | Name | Description |
| PUB | Public | Unauthenticated Users |
| USR | User | Authenticated Users |
| OWN | Owner | Content Owner (profile, post, comment) |
| SM | System Manager | System Managers |

**7.3. OpenAI Specification**

**A8: Vertical prototype**

**8.1. Implemented Features**

**8.1.1. Implemented User Stories**

|  |  |  |  |
| --- | --- | --- | --- |
| Identifier | Name | Priority | Description |
| US101 | View Top News Feed | High | As a User, I want to view a feed of top news, so I can quickly access the most relevant or popular articles. |
| US102 | View Recent News Feed | High | As a User, I want to view a feed with the latest news, so that I can keep up with the latest updates and news. |
| US103 | View News Item | High | As a User, I want to view the full content of a news item, so that I can read the story in detail. |
| US104 | View News Item Comments | High | As a User, I want to view other users comments on a news story, so that I can see their opinions and interact with the content more fully. |
| US105 | Search for News Items and Comments | High | As a User, I want to search for news and reviews, so I can find specific information or topics of interest. |
| US202 | View News Item Details | High | As an Authenticated User, I want to view the full details of a news item, so that I can get more information about the author, publication date, votes, and other metadata. |
| US203 | View Comment Details | High | As an Authenticated User, I want to view the full details of a comment on a news item, so that I can know who made the comment, when it was posted, and how many votes it has received. |
| US204 | Create News Item | High | As an Authenticated User, I want to create and publish a news item, so that I can share relevant information or content with other users. |
| US205 | Vote on News Item | High | As an Authenticated User, I want to be able to vote on news items so that I can express my opinion about the content and influence its visibility. |
| US207 | Comment on News Item | High | As an Authenticated User, I want to comment on a news item, so that I can participate in discussions and share my opinions. |
| US208 | View Other Users' Profiles | High | As an Authenticated User, I want to view other users' profiles, so that I can see their activities, reputation, and published content. |
| US209 | View Reputation of Other Users | High | As an Authenticated User, I want to view the reputation of other users, so that I can assess their credibility and influence within the community. |
| US210 | View News Items Tags | High | As an Authenticated User, I want to view the tags associated with a news item, so that I can better understand the context and related topics of the content. |
| US214 | View My Profile | High | As an Authenticated User, I want to view my own profile, so that I can see my activity, reputation, and published content. |
| US215 | Edit My Profile | High | As an Authenticated User, I want to edit my own profile, so that I can change my information and profile picture. |
| US217 | Logout | High | As an Authenticated User, I want to logout of my account, so that I can become un unauthenticated user. |
| US301 | Edit News Item | High | As a News Author, I want to be able to edit an existing news item so that I can correct information or update content according to new data or events. |
| US401 | Edit Comment | High | As a Comment Author, I want to edit a comment I posted, so that I can correct errors or adjust the content according to new information or points of view. |
| US501 | Sign-in | High | As a Unauthenticated User, I want to authenticate into the system, so that I can access privileged information. |
| US502 | Sign-up | High | As a Unauthenticated User, I want to register myself into the system, so that I can access privileged information. |

Table 70

**8.1.2. Implemented Web Resources**

**- Module M01: Authentication**

|  |  |
| --- | --- |
| Web Resource Reference | URL |
| R101: Register Form | GET /register |
| R102: Register a new user | POST /register |
| R103: Login Form | GET /login |
| R104: User login | POST /login |
| R105: Logout user | GET /logout |

Table 71

**- Module M02: Users**

|  |  |
| --- | --- |
| Web Resource Reference | URL |
| R201: Get User Profile | GET /users/{id} |
| R203: Edit Profile Form | GET /users/{id}/edit |
| R204: Update User Profile | PUT /api/users/{id}/edit |

Table 72

**- Module M03: News Posts**

|  |  |
| --- | --- |
| Web Resource Reference | URL |
| R301: Create a New Post | GET /posts |
| R301: Create a New Post (API) | POST /api/posts |
| R304: Edit a Post | GET /posts/{post\_id} |
| R304: Edit a Post (API) | GET /posts/category/{category\_id} |
| R307: Vote on a Post (API) | POST /api/posts/{post\_id}/vote |
| R308: Edit Vote on a Post (API) | PUT /api/posts/{post\_id}/vote |
| R309: Remove Vote on a Post (API | DELETE /api/posts/{post\_id}/vote |
| R314: Create Post Form | GET /posts/create |
| R315: Edit Post Form | GET /posts/{post\_id}/edit |

Table 73

**- Module M04: Comments**

|  |  |
| --- | --- |
| Web Resource Reference | URL |
| R401: Add a Comment to a Post | GET /users/{id} |
| R403: Edit a Comment | GET /users/{id}/edit |
| R406: Vote on a Comment | PUT /api/users/{id}/edit |
| R407: Edit a Vote on a Comment | PUT /api/comments/{comment\_id}/vote |
| R408: Remove Vote on a Comment | DELETE /api/comments/{comment\_id}/vote |
| R410: Edit Comment Form | GET /comments/{comment\_id}/edit |

Table 74

**- Module M05: Search**

|  |  |
| --- | --- |
| Web Resource Reference | URL |
| R501: Search Posts by Keywords | GET /search/posts |
| R503: Search Users by Username or Email | GET /search/users |

Table 75

**8.2. Prototype**

Our objective was to establish the core features of the project, ensuring functional robustness and user-focused utility. While the visual design has not been our main focus and thus remains in a basic state, it effectively demonstrates the overall layout and facilitates straightforward navigation throughout the website.

**A9: Product**

**9.1. Installation**

- Docker commands to start the latest image

**docker run -d --name lbaw2484 -p 8001:80 gitlab.up.pt:5050/lbaw/lbaw2425/lbaw2484**

**9.2. Usage**

**9.2.1. Administration Credentials**

|  |  |
| --- | --- |
| Username | Password |
| [admin@lbaw2484.com](mailto:admin@lbaw2484.com) | 12345678 |

Table 76

**9.2.2. User Credentials**

|  |  |  |
| --- | --- | --- |
| Type | Email | Password |
| Authenticated User | cristiano\_cr7\_ronaldo@goat.pt | 12345678 |

Table 77

**9.3. Application Help**

Additionally, action icons are equipped with titles that clearly describe their functionality, helping users understand the purpose of each action. We also use alert messages and error notifications throughout the application to guide users when something goes wrong, ensuring they can quickly identify and address issues.

**9.4. Input Validation**

<label for="image">News Cover Image:</label>

<input

type="file"

id="news-image"

name="image"

class="form-control"

accept="image/\*"

>

@error('image')

<div class="error text-danger">{{ $message }}</div>

@enderror

$request->validate([

'title' => 'required|string|max:255',

'synopsis' => 'nullable|string|max:300',

'body' => 'required|string|max:10000',

'categories' => 'required|array',

'categories.\*' => 'exists:categories,category\_id',

'image' => 'nullable|image|mimes:jpeg,png,jpg,gif',

]);

**9.5. Check Accessibility and Usability**

Accessibility: https://gitlab.up.pt/lbaw/lbaw2425/lbaw2484/-/blob/main/checklists/Checklist\_de\_Acessibilidade-SAPO\_UX.pdf

Usability: https://gitlab.up.pt/lbaw/lbaw2425/lbaw2484/-/blob/main/checklists/Checklist\_de\_Usabilidade-SAPO\_UX.pdf

**9.6. HTML & CSS Validation**

HTML: https://gitlab.up.pt/lbaw/lbaw2425/lbaw2484/-/blob/main/checklists/Showing\_results\_for\_contents\_of\_text-input\_area-Nu\_Html\_Checker.pdf

CSS: https://gitlab.up.pt/lbaw/lbaw2425/lbaw2484/-/blob/main/checklists/Resultados\_da\_valida%C3%A7%C3%A3o\_CSS\_do\_W3C\_para\_app.css\_(CSS\_n%C3%ADvel\_3+SVG).pdf

**9.7. Revisions to thhe Project**

**9.8. Implementation Details**

**9.8.1. User Credentials**

**9.8.2. User Stories**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| US Identifier | Name | Module | Priority | Team Members | State |
| US501 | Sign-in | M01 | High | Carlota Leite | 100% |
| US502 | Sign-up | M01 | High | Carlota Leite | 100% |
| US217 | Logout | M01 | High | Carlota Leite | 100% |
| US102 | View Recent News Feed | M03 | High | Alexandre Santos | 100% |
| US103 | View News Item | M03 | High | Alexandre Santos, Carlota Leite | 100% |
| US202 | View News Item Details | M03 | High | Alexandre Santos | 100% |
| US214 | View My Profile | M02 | High | Alexandre Santos, Eduarda Araújo | 100% |
| US208 | View Other Users' Profiles | M02 | High | Alexandre Santos | 100% |
| US210 | View News Items Tags | M03 | High | Alexandre Santos | 100% |
| US204 | Create News Item | M03 | High | Alexandre Santos | 100% |
| US104 | View News Item Comments | M03 | High | Alexandre Santos | 100% |
| US203 | View Comment Details | M03 | High | Alexandre Santos | 100% |
| US207 | Comment on News Item | M04 | High | Alexandre Santos | 100% |
| US301 | Edit News Item | M03 | High | Carlota Leite, Alexandre Santos | 100% |
| US401 | Edit Comment | M04 | High | Alexandre Santos | 100% |
| US215 | Edit My Profile | M02 | High | Alexandre Santos, Eduarda Araújo | 100% |
| US101 | View Top News Feed | M03 | High | Alexandre Santos, Carlota Leite | 100% |
| US205 | Vote on News Item | M03 | High | Alexandre Santos | 100% |
| US105 | Search for News Items and Comments | M05 | High | Alexandre Santos, Carlota Leite | 100% |
| US209 | View Reputation of Other Users | M02 | High | Alexandre Santos | 100% |
| US107 | About Us | M08 | High | Carlota Leite | 100% |
| US108 | Contacts | M08 | High | Carlota Leite | 100% |
| US206 | Vote on Comment | M04 | High | Alexandre Santos | 100% |
| US218 | Add/Remove News Item from Favorites | M03 | Medium | Alexandre Santos | 100% |
| US219 | Report Users or Content | M02 | Medium | Alexandre Santos, Carlota Leite, João Couto | 100% |
| US213 | Follow Notification | M07 | High | Alexandre Santos, João Couto | 100% |
| US303 | Vote Notification | M07 | High | Alexandre Santos, João Couto | 100% |
| US304 | Comment Notification | M07 | High | Alexandre Santos, João Couto | 100% |
| US403 | Vote Notification | M07 | High | Alexandre Santos, João Couto | 100% |
| US404 | Comment Notification | M07 | High | Alexandre Santos, João Couto | 100% |
| US211 | Follow/Unfollow Users | M02 | High | Alexandre Santos | 100% |
| US212 | Follow/Unfollow Tags | M02 | High | Alexandre Santos | 100% |
| US603 | Block and Unblock User Accounts | M06 | High | Carlota Leite, Alexandre Santos | 100% |
| US608 | Manage Users or Content Reports | M06 | Low | Alexandre Santos, Carlota Leite | 100% |
| US201 | View User News Feed | M03 | High | Alexandre Santos | 100% |
| US602 | Administer User Accounts | M06 | High | Alexandre Santos | 100% |
| US606 | Delete Comments | M06 | Medium | Carlota Leite, Alexandre Santos, Eduarda Araújo | 90% |
| US216 | Delete Account | M02 | High | Eduarda Araújo, Alexandre Santos | 99% |
| US604 | Delete User Accounts | M06 | High | Eduarda Araújo, Alexandre Santos | 99% |
| US302 | Delete News Item | M03 | High | Alexandre Santos, Eduarda Araújo | 50% |
| US402 | Delete Comment | M04 | High | Alexandre Santos, Eduarda Araújo | 50% |
| US503 | Recover Password | M01 | Low | - | 0% |
| US504 | OAuth API Sign-in | M01 | Low | - | 0% |
| US505 | OAuth API Sign-up | M01 | Low | - | 0% |
| US106 | Help | M08 | Low | - | 0% |
| US601 | Manage Administrator Accounts | M06 | Low | - | 0% |
| US607 | Statistics | M06 | Low | - | 0% |
| US220 | Propose new Topic | M02 | Low | - | 0% |
| US605 | Manage Topic Proposals | M06 | Low | - | 0% |
| US221 | Appeal for Unblock | M02 | Low | - | 0% |
| US609 | Manage Unblock Appeals | M06 | Low | - | 0% |

Table 78

**A10: Presentation**

**10.1. Product presentation**

**10.2. Video presentation**