**1) Introduction**

***a) Purpose***

The purpose of this document is to define the software requirements for a multi-lingual open online encyclopedia. This application allows a community of volunteers to collaboratively manage articles in different languages, with some features including but not limited tino browsing and bookmarking articles, customizing feed (by choosing articles categories), and search history for articles.

**b) Project Scope**

- Enable user registration and authentication

- Allow article creation, editing, and version tracking

- Include content moderation and approval workflows

- Support search, categorization, and tagging of articles

- Ensure responsive design and accessibility compliance

**c) Glossary and Abbreviations**

|  |  |
| --- | --- |
| CRUD | Create, Read, Update, and Delete |
| DB | Database |

**d) List of the System Stakeholders**

- Investors

- Authors

**e) References**

- Wikipedia.com

- YouTube.com

**2) Functional Requirements**

**(a, b) Requirements Specification**

1. Users shall be able to log in and register.

1.1 The system shall allow new users to register using valid credentials.

1.2 The system shall validate the uniqueness of usernames.

1.3 The system shall provide error messages for incorrect credentials.

1.4 The system shall confirm the login (if it is successful) and redirect the user to the homepage.

2. Users shall be able to customize their explore feed.

2.1 The system shall display a random article page by default until the user customizes the

articles used in their favorite categories.

2.2 The system shall provide the user with an option to choose a favorite category.

2.3 The system shall generate a customized explore feed based on the user’s chosen category.

3. Users shall be able to search their history

3.1 The system shall store the search history for each user in the DB.

3.2 The system shall allow users to view a list of previously searched articles.

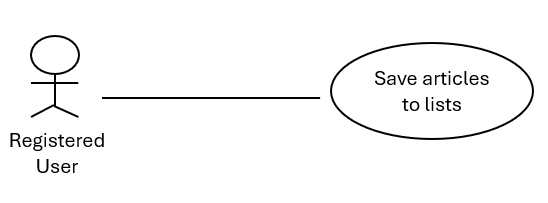
4. Editors shall be able to manage [CRUD] articles.

4.1 The system shall allow editors to create new articles.

4.2 The system shall allow editors to edit existing articles.

5. Registered users shall be able to save articles to their created lists.

5.1. The system shall save the selected article to the selected list.

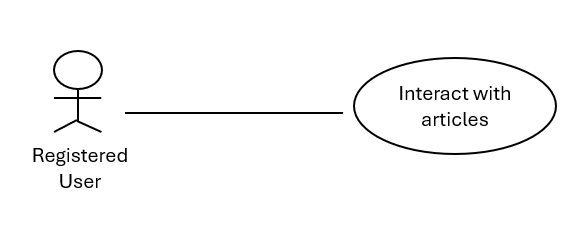


 6. Registered users shall be able to bookmark articles.

  6.1) The system shall identify the articles that the registered user wants to bookmark.  
  6.2) The system shall be able to save the articles (URL and title) to the lists.

**7**. Registered users can interact with articles

**7.1** Each article should have an option for liking and comments section.

****

**8**. Users can open the “trending” section

**8.1** Each week, the system shall look up the top 3 articles (based on upvotes number) and put them in the trending section.

9. Users can listen to articles.

9.1 The system should have a speaker button to read the article for the user.

10. The system shall display a random feed if no categories chosen

10.1 The system shall select random articles from the DB.

10.2 The system shall provide the user with an interface to display the selected articles from the DB.

10.3 Registered Users should be able to select preferred categories from the randomly displayed articles.

11. Admins shall be able to generate a pdf report of the system stats  
11.1) The system shall be able to generate a pdf report of the system stats including users number and articles number.

12. Users shall be able to view articles in multiple languages.

|  |  |
| --- | --- |
| Function | Changes the article’s language. |
| Description | Function enhances the system usability of the system by making articles multilingual. |
| Inputs | Language you want the article to be translated to. |
| Source | User. |
| Outputs | Translated article |

13) The admin shall be able to remove articles.  
13.1) Selected article to be removed shall be removed.

14) The users' passwords shall be saved **hashed.**

14.1) The system saves users’ passwords hashed in the DB.

15) Users can report articles to be sent to admin

15.1) Report button for users

16) Admins shall be able to update user roles

16.1) Update role button

17) Admins shall be able to delete users

17.1) Delete user button

18) Users shall be able to generate random article

18.1) Generate random article button

19) Admins shall be able to delete comments

19.1) Delete article button for every comment for admins

**(c) Requirements Priorities (using Moscow Scheme)**

**Must have**

* User registration and login
* Article creation and editing
* View articles in multiple languages
* Searching for articles

**Should have**

* Bookmarking articles
* Organizing articles into custom lists
* Offline access to save lists
* Feed customization

**Could have**

* Share articles with other users
* Tagging other users on articles
* Different themes

**Will not have**

* Chat among users

**3) Non-functional Requirements**

**a) General Types:**

- Performance

- Usability

- Look & Feel

- Operational & Environmental

**b) Non-functional requirements Specification**

1- The system must be mobile responsive (Enviromental and Operational).

2-The look and feel of the system shall be clean and user friendly. It shall use an eye

colors like white, green, and black. The font size shall be well readable. And provides a

smooth experience by using lists and navigations across different devices (Look and Feel).

3-The system shall have a simple interface and be easy to navigate for both contributors and readers (Usability).

4- The system shall enable the user to switch between multiple articles languages (Usability).

5- The System shall load the article pages within 5 seconds (Performance).

**c) Fit criteria for non-functional requirements**

1- The system’s UI must adapt and function correctly on screens ranging from 320px (phones) to 1920px (monitors).

2- After accessing the website, you can see that consistency of fonts and colors across all

pages are maintained. By clicking on the lists and navigation bars you can see the smooth

and user-friendly usage of the system, also, all text is displayed using the selected font size.

header, footer, and navigations appear in the same location on all pages.

3- Users would be able to understand how to use the system within 5 minutes.

4- "When changing the language, the user should remain at the same reading position in the article, if possible.

**d) System’s overall Architecture Affection**

1- The system now can be accessed easily from the phone user do not need PC or laptop.

2- It will make the usage of the system more user-friendly, comfortable for the user, and easier to use. It ensures a responsive, consistent, and scalable interface, allowing users to directly Contribute info and experience a smoother and more elegant user experience. While streamlining the system to maintain a clean, user-friendly look across all pages

3- An efficient API design and performance optimization at both frontend and backend levels.

4- Users from various parts of the world can read the system’s article.

5- The user able to interact with system quickly and in real time .in the case of any changes or update to articles these changes should be immediately reflect to the user.

**4) Design & Implementation Constraints**

- The system shall fully support right-to-left languages such as Arabic, which affects UI design and layout.

- The system shall be integrated within an existing translation API.

- All user data must be handled according to compliance and privacy standards.

**5) System Evolution**

**a)** We might add sharing articles feature with users, and the ability to tag other users on articles.

**b)** Anticipated future changes should guide system design to be **flexible and scalable**. This means building the system in a way that allows parts to be easily updated, replaced, or expanded without affecting the entire system.

**6) Requirements Discovery Approaches Relied On**

* **Document analysis:** looking at how Wikipedia manages multilingual content.
* **Use cases/User stories:** helped us to understand and identify needed features.
* **Brainstorming sessions:** by exchanging information between our team members.
* **Observation:** watch how users currently do tasks and see how people use existing online encyclopedias or translation tools.
* **AI services:** some AI services that helped us gather some information.

**7) Requirements Validation Techniques That Will Be Used**

1. **Prototyping**

**1.1 Description**

The prototype of the system is presented to the end-user or customer, they experiment with the presented model and check if it meets their need.

**1.2 Goal**

Clarify unclear requirements, especially for UI/UX or workflows.

**1.3 Help with**

User feedback, refining ambiguous requirements.

**2. Requirements Reviews**

**2.1 Description**

In this approach, the SRS is carefully reviewed by a group of people including

both the contractor organization and the client side, the reviewer systematically

analysis the document to check errors.

**2.2 Goal**

Identify errors, ambiguities, and inconsistencies.

**2.3 Types**

2.3.1 Informal reviews: Unstructured walkthroughs.

2.3.2 Formal reviews: Structured, documented, and planned meetings.

--------------------------------------------------------------------------------------------------------------------------------------------