Software Requirements Specification

for

**"Insight Wave: News Nexus"**

**Version 1.0 approved**

**Prepared by Shruti Pathak(E23MCAG0033), Aryan Sood(E23MCAG0047)**

**Bennett University**

**11-03-2024**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. System Features 3**

3.1 System Feature 1 3

3.2 System Feature 2 (and so on) 4

**4. External Interface Requirements 4**

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 4

4.4 Communications Interfaces 4

**5. Other Nonfunctional Requirements 5**

5.1 Performance Requirements 5

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

**6. Other Requirements 5**

**Appendix A: Data Flow Diagram 5**

# Introduction

## Purpose

*InsightWave: News Nexus tailors your daily news feed based on your interests and preferences. Using advanced algorithms, it not only delivers the latest headlines but also gauges the emotional tone of each article. Through this innovative approach, InsightWave aims to empower users with a more informed, balanced, and personalized news experience, enhancing their ability to navigate and comprehend the complex world of current affairs.*

## Document Conventions

*Bold Text: Section headings*

*Italic Text: Emphasis or placeholder text*

## Intended Audience and Reading Suggestions

*This document is intended for the project stakeholders, including developers, designers, testers, and project managers. It provides a comprehensive understanding of the system requirements and serves as a reference throughout the development process.*

## Project Scope

*The project aims to develop a personalized news aggregator software that collects, curates, and delivers news articles based on user preferences and interests. The software will utilize AI algorithms to analyze user behavior and provide tailored news recommendations.*

## References

*API Key: d2ee0448c8314fd4bb69e71a9fd145f1*

[*https://polymer-library.polymer-project.org/1.0/docs/apps/news-case-study*](https://polymer-library.polymer-project.org/1.0/docs/apps/news-case-study)

[*https://creately.com/diagram/example/ict7ohgh1/next-level-news-application-classic*](https://creately.com/diagram/example/ict7ohgh1/next-level-news-application-classic)

*https://www.scribd.com/document/442587699/News-Portal-Project-report*

# Overall Description

## Product Perspective

### *Introduction to the Industry Context:*

* *Provide an overview of the industry or sector in which the software operates (e.g., news media, information technology).*
* *Describe key trends, challenges, and opportunities shaping the industry landscape.*

### *Positioning within the Ecosystem:*

* *Identify the stakeholders and entities that interact with or are impacted by the software.*
* *Explain the role of the software within the larger ecosystem, including how it addresses specific needs or pain points of stakeholders.*

### *Integration with Existing Systems or Platforms:*

* *Outline any existing systems, platforms, or technologies that the software integrates with or complements.*
* *Describe the interoperability requirements and protocols for seamless integration with other systems.*

### *Competitive Analysis:*

* *Conduct a competitive analysis to identify similar solutions or competitors within the industry.*
* *Highlight the unique features, advantages, and value propositions that differentiate the software from competitors.*

### *Industry Standards and Regulations:*

* *Discuss relevant industry standards, guidelines, or regulations that the software must comply with (e.g., data privacy regulations, industry best practices).*
* *Describe how the software ensures compliance with these standards and regulations.*

### *Future Trends and Evolution:*

* *Anticipate future trends, innovations, and developments within the industry that may impact the software's evolution.*
* *Discuss potential scalability, adaptability, and future-proofing measures to accommodate changes in the industry landscape.*

## Product Features

**User Profile Creation**: Allow users to create personalized profiles where they can specify their interests, preferences, and favorite topics.

**Content Customization**: Enable users to customize their news feed by selecting specific topics, sources, keywords, or regions they want to follow.

**AI-Powered Recommendation Engine**: Utilize machine learning algorithms and AI-driven recommendation engines to analyze user behavior, preferences, and engagement patterns to suggest relevant news articles.

**Topic Categorization:** Categorize news articles into different topics or categories (e.g., politics, technology, sports, entertainment) for easier navigation and filtering.

**Real-Time Updates:** Provide real-time updates and notifications for breaking news, trending topics, and personalized recommendations based on user preferences.

**Multi-Source Aggregation:** Aggregate news content from a variety of sources, including reputable news websites, blogs, RSS feeds, social media platforms, and user-generated content.

**Content Filtering:** This allows users to filter news content based on factors such as relevance, recency, popularity, and source credibility.

## User Classes and Characteristics

**General Users:** These are regular users who consume news content through the aggregator. They have varying interests and preferences, and they rely on the platform to provide personalized news recommendations.

**Premium Users**: Premium users may have access to additional features or premium content. They may pay for a subscription or premium membership to unlock exclusive content, ad-free browsing, or other premium features.

**Contributors:** Contributors are users who actively contribute content to the platform. They may submit articles, blog posts, or other user-generated content for inclusion in the aggregator.

**Administrators:** Administrators are responsible for managing and overseeing the operation of the news aggregator platform. They have access to administrative features such as user management, content moderation, and platform customization.

**Content Providers**: Content providers are organizations or individuals that supply news content to be included in the aggregator. They may have partnerships or agreements with the aggregator to distribute their content to a wider audience.

**Advertisers**: Advertisers are entities that place advertisements on the news aggregator platform. They may target specific user segments or demographics based on their advertising goals.

**Analytics Users**: Analytics users are interested in analyzing user behavior, engagement metrics, and other data related to the platform's performance. They may include data analysts, marketers, or business intelligence professionals.

**Developers**: Developers are individuals or teams responsible for building, maintaining, and updating the news aggregator software. They may develop new features, fix bugs, and ensure the platform's technical performance.

## Operating Environment

* **Web Platform:**
  + The personalized news aggregator software will be deployed as a web application accessible via standard web browsers.
  + Supported web browsers may include Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, and others.
  + The software should be compatible with both desktop and mobile web browsers to accommodate users accessing the platform from various devices.
* **Mobile Applications:**
  + Native mobile applications will be developed for iOS and Android platforms to provide a seamless user experience on mobile devices.
  + The iOS app will be compatible with devices running iOS version X and above, while the Android app will support devices running Android version Y and above.
  + The mobile apps will leverage platform-specific features and design guidelines to optimize performance and user interface for each platform.
* **Desktop Applications:**
  + A desktop application may be developed for Windows and macOS platforms to offer a dedicated experience for desktop users.
  + The desktop application will be compatible with Windows 10 and above, as well as macOS X and above.
  + It will provide similar functionality to the web and mobile versions of the software but optimized for desktop usage.

## Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## User Documentation

* **Data Privacy and Security Regulations:**
  + The software must comply with data privacy regulations such as GDPR, CCPA, and other relevant laws.
  + User data collected by the news aggregator must be handled securely, with appropriate measures in place to protect against unauthorized access, data breaches, and misuse.
* **Content Licensing and Copyright Restrictions:**
  + The content displayed on the news aggregator may be subject to licensing agreements and copyright restrictions.
  + The software must respect intellectual property rights and adhere to content usage policies outlined by content providers and publishers.
* **Third-Party API Limitations:**
  + The news aggregator may rely on third-party APIs to access news sources, social media platforms, or other external services.
  + The availability, reliability, and usage limits of these APIs may impose constraints on the software's functionality and performance.
* **Platform Compatibility:**
  + The software must be compatible with a variety of web browsers, operating systems, and devices.
  + Design and development decisions must consider compatibility requirements to ensure a consistent and optimal user experience across different platforms and devices.
* **Scalability and Performance:**
  + As the user base and content volume grow, the software must be scalable to accommodate increased traffic and data processing requirements.
  + Performance optimizations may be necessary to maintain responsiveness and reliability under high load conditions.
* **Resource Constraints:**
  + Budgetary constraints may limit the availability of resources for the development, testing, and deployment of the software.
  + Development timelines and project scope may need to be adjusted to align with resource limitations and priorities.
* **Technical Limitations:**
  + Technical constraints such as hardware limitations, network connectivity issues, and compatibility issues with legacy systems may impact the design and implementation of the software.
  + The software architecture and technology stack must be chosen carefully to mitigate technical risks and constraints.

## Assumptions and Dependencies

**Assumptions:**

Users will actively engage with the personalized news aggregator by customizing their preferences and regularly accessing the platform. The success of the aggregator relies

on user engagement and adoption of personalized features.Sufficient and diverse content

will be available from news sources, blogs, and other content providers for aggregation.

Users will consent to sharing their data and preferences to personalize their news feeds. The software will be compatible with mainstream web browsers (e.g., Chrome, Firefox, Safari) and mobile operating systems (e.g., iOS, Android).

**Dependencies:**

* **API Integrations:** The functionality of the personalized news aggregator relies on integrating with third-party APIs to access news sources, social media platforms, and other external services.
* **Content Partnerships:** Establishing partnerships with content providers, publishers, and news outlets is necessary to access and aggregate high-quality news content.
* **Data Storage and Processing Infrastructure:** Adequate infrastructure for data storage and processing is required to handle the volume of user data and content metadata generated by the aggregator.
* **Compliance with Regulations:** Compliance with data privacy regulations (e.g., GDPR, CCPA) and copyright laws is necessary to ensure the legal operation of the news aggregator.

# System Features

## System Feature 1

***User Profile Creation and Summarisation:***

**Description and Priority:**

The software will allow users to create personalized profiles where they can specify their interests, preferences, and favorite topics. This feature enables the customization of the news feed to match individual user preferences, enhancing the user experience and relevance of the content delivered.

* ***User Registration****: Users should be able to register for an account by providing basic information such as name, email address, and password.*
* ***Profile Creation:*** *Upon registration, users can create personalized profiles by providing additional details such as age, gender, location, and interests.*
* ***Interest Selection:*** *Users can select topics, categories, or keywords of interest from predefined lists or by entering custom preferences.*
* ***Preference Management****: Users should have the ability to modify, update, or remove their interests and preferences at any time.*
* ***User Authentication:*** *Secure authentication mechanisms should be implemented to verify user identity and protect user profiles from unauthorized access.*
* ***Personalization Algorithms:*** *The software should employ algorithms to analyze user preferences and behavior, recommending personalized content based on their profile settings and interactions.*
* ***Privacy Controls:*** *Users should have control over their privacy settings, including options to manage data-sharing preferences and opt out of personalized recommendations if desired.*

## Additional System Features:

**Advanced Content Filtering and Sorting**: This allows users to filter and sort news content based on various criteria such as relevance, recency, source credibility, and popularity. Provide advanced filtering options including by topic, keyword, author, publication date, and sentiment.

**Bookmarking and Saved Articles**: Enable users to bookmark or save articles for later reading or reference. Implement a dedicated "Saved Articles" section where users can access their bookmarked content across devices.

**Notification and Alert System**: Offer customizable notification settings for users to receive alerts about breaking news, trending topics, or updates related to their interests. Allow users to configure notification preferences such as frequency, content type, and delivery method (e.g., email, push notification).

**Social Media Integration**: Integrate social media sharing functionality to allow users to share interesting articles with their social networks. Provide options for users to connect their social media accounts to the news aggregator for personalized recommendations and social sharing features.

**Multilingual Support**: Support multiple languages to cater to a diverse user base and provide news content in users' preferred languages. Implement language detection and automatic translation features to deliver content in users' preferred languages.

**Readability and Accessibility Enhancements**: Offer readability enhancements such as font adjustment, night mode, and text-to-speech functionality to improve accessibility for users with visual impairments. Ensure compliance with web accessibility standards (e.g., WCAG) to make the software accessible to users with disabilities.

**Content Discovery and Exploration:** Provide features for users to discover new content and explore topics of interest beyond their predefined preferences. Implement discovery features such as related articles, recommended topics, and trending stories to engage users and encourage exploration.

**Community Engagement and Interactivity**: Foster community engagement by allowing users to comment on articles, participate in discussions, and interact with other users. Implement features for users to like, share, and react to articles, fostering a sense of community and social interaction within the platform.

**Offline Reading and Synchronization**: Enable users to access and read saved articles offline, allowing them to continue reading even when they are not connected to the internet. Implement synchronization features to ensure that user preferences, saved articles, and reading progress are synchronized across devices.

**Premium Subscription Options:** Offer premium subscription tiers with additional features such as ad-free browsing, exclusive content, personalized newsletters, and premium support. Provide flexible subscription plans with options for monthly, annual, or customized billing cycles.

# External Interface Requirements

## User Interfaces

***Layout design:***

***Responsive Design****: The user interface should be responsive and adapt to various screen sizes and resolutions, ensuring a consistent user experience across devices.*

***Whitespace:*** *Use whitespace effectively to improve readability, reduce clutter, and create a visually appealing interface.*

***Content Presentation:*** *Present news articles in a visually engaging manner, with clear headlines, images, and summaries to attract user attention.*

#### Navigation Design:

***Intuitive Navigation****: Design intuitive navigation menus and controls to facilitate easy access to different sections and features of the software.*

***Hierarchy:*** *Establish a clear hierarchy of navigation elements, with primary navigation options easily accessible and secondary options available through dropdown menus or submenus.*

***Search Functionality:*** *Include a prominent search bar or search icon for users to quickly search for specific topics, keywords, or articles within the app.*

#### Visual Design:

#### *Consistent Branding: Maintain consistent branding elements such as colors, fonts, and logo placement throughout the user interface to reinforce brand identity.*

***Visual Hierarchy:*** *Use contrast, color, and typography techniques to emphasize important elements and guide user attention.*

## Hardware Interfaces

High-performance server infrastructure with sufficient processing power, memory, and storage capacity to handle user requests, data processing, and content delivery.

Support for scalable cloud hosting services such as AWS, Google Cloud Platform, or Microsoft Azure to accommodate growing user traffic and data volume.

Standard desktop computers, laptops, tablets, and smartphones with modern web browsers (e.g., Google Chrome, Mozilla Firefox, Safari, Microsoft Edge) for accessing the web-based software.

Compatibility with various operating systems including Windows, macOS, Linux, iOS, and Android.

## Software Interfaces

* News API (e.g., NewsAPI.org)
* RSS feeds from reputable news outlets and publishers
* requests
* TextBlob

## External Interface Requirments Use Case/Scenario

**Use Case 1: Stay Informed with Positive News**

**User Story:**

* **As a busy professional, I want to read news articles about my field that have a positive tone. This way, I can stay informed without feeling overwhelmed by negative news.**

**Use Case Description:**

1. The user enters their field of interest (e.g., "technology").
2. The system calls the news API to retrieve articles about that topic.
3. The system analyzes the sentiment of each article.
4. The system recommends articles with positive sentiment to the user.

**Use Case 2: Personalized News Exploration**

* **As a curious student, I want to explore news articles about diverse topics and receive recommendations based on my reading preferences. This way, I can expand my knowledge and discover new interests.**

**Use Case Description:**

1. The user either specifies a topic or chooses to explore general news.
2. The system retrieves a set of articles from the news API.
3. The user reads some of the articles.
4. The system discreetly analyzes the sentiment of those articles.
5. The system recommends further articles with a similar sentiment to the user's initial readings.

**Use Case 3: Curated Content for Research or Mood-Boosting**

**User Story:**

* **As a researcher, I want to compile articles with a specific sentiment for my research study. This tool can help me quickly gather relevant data.**
* **As someone who struggles with mood swings, I want to read news articles with a positive tone to help lift my spirits.**

**Use Case Description:**

1. The user specifies the desired sentiment (positive, negative, or neutral).
2. The system retrieves articles on a chosen topic or a general news feed.
3. The system analyzes the sentiment of each article.
4. The system recommends articles that match the user's specified sentiment.

**Additional Considerations:**

* The code could be integrated into a larger news aggregation platform for a more comprehensive user experience.
* It could be adapted for different domains, such as product reviews or social media sentiment analysis.
* The ethical implications of sentiment analysis and recommendation algorithms should be carefully considered.

# Other Nonfunctional Requirements

**Performance**: User profile creation and personalization algorithms should operate efficiently, with minimal latency, to deliver a seamless user experience.

**Scalability:** The system should be capable of handling a growing user base and increasing data volume without sacrificing performance or responsiveness.

**Security**: User profiles and personal data should be securely stored and protected against unauthorized access, data breaches, and malicious activities.

**Usability**: The user interface for profile creation and management should be intuitive, user-friendly, and accessible across different devices and platforms.

**Compliance**: The software should comply with relevant data privacy regulations (e.g., GDPR, CCPA) and industry standards for user data protection and privacy.

## Software Quality Attributes

**Reliability:**

Reliability ensures consistent performance and accurate content delivery. Fault tolerance mechanisms will be implemented to handle errors and system failures gracefully, minimizing downtime and ensuring uninterrupted service. Robust error handling procedures will detect, report, and recover from errors efficiently, maintaining the reliability of the platform. Data integrity measures, including validation and backup procedures, will safeguard against data corruption, ensuring the accuracy and reliability of stored information. Ultimately, reliability instills trust in users, assuring them of a dependable news aggregation platform.

**Maintainability:**

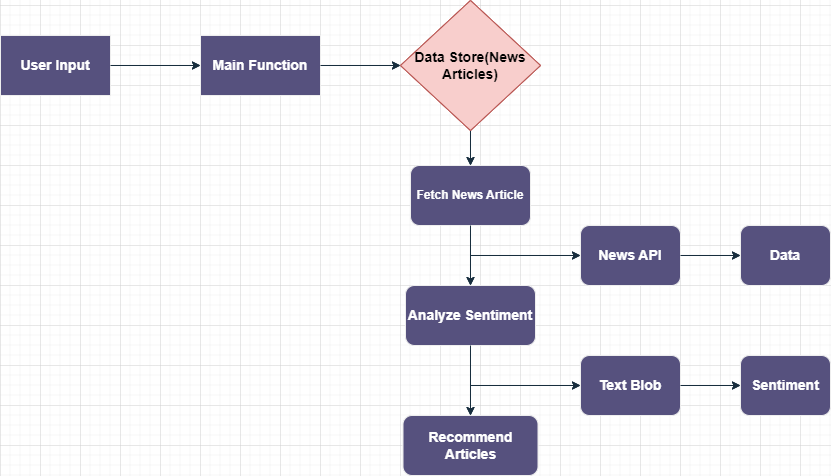
The modular architecture will facilitate easy modification and enhancement of individual components, ensuring flexibility and scalability. Comprehensive documentation will provide developers and administrators with the necessary guidance for understanding and maintaining the platform effectively.

**Scalability:**

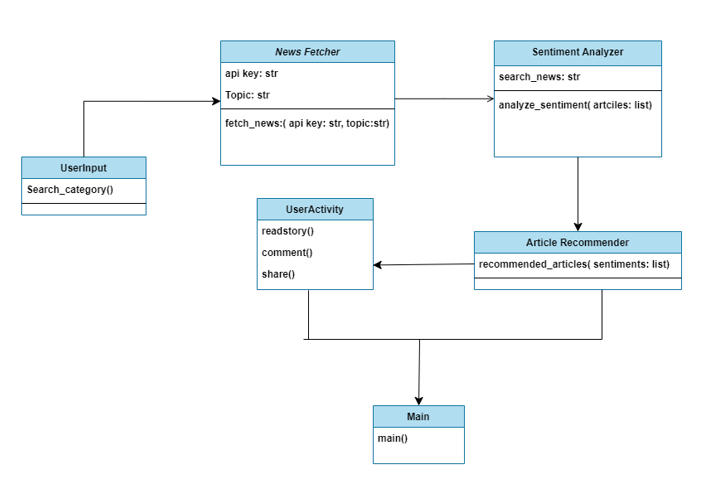
Horizontal scaling mechanisms will allow the platform to distribute workload and increase capacity by adding more resources as needed. Vertical scaling through hardware upgrades will ensure the platform can handle the increased demand on individual servers or instances. Load balancing mechanisms will evenly distribute incoming traffic across multiple servers, optimizing performance and resource utilization. Additionally, caching strategies will reduce latency and improve scalability by storing frequently accessed content closer to users. Scalability ensures the personalized news aggregator can adapt to changing demands and support growth effectively.

**Appendix A: Diagram**

**Data flow diagram-**

**

**Class diagram-**

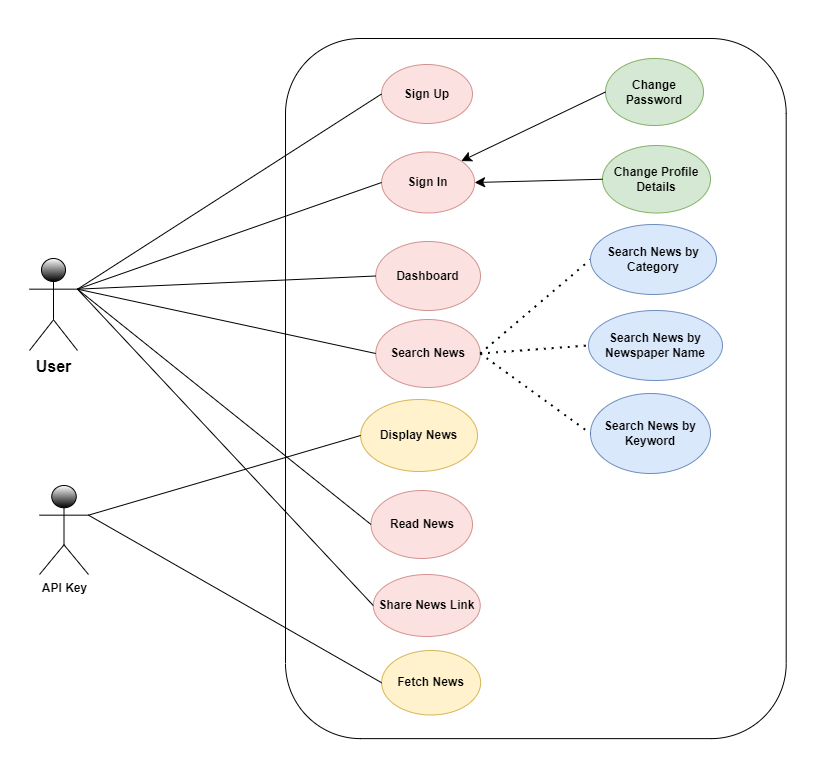
****

**Description-**

In a class diagram, a class serves as the foundational building block, embodying a specific concept, entity, or blueprint for generating objects. It contains both data (attributes) and behaviors (methods) that define the characteristics and functionalities of the object.

1. **Top section**: Identifies the class name, providing a clear label for the entity it represents.
2. **Middle section**: Enumerates the attributes of the class, outlining the various properties or characteristics associated with instances of the class.
3. **Bottom section**: Lists the methods or functions associated with the class, detailing the operations or behaviors that instances of the class can perform.

**Use Case Diagram**-



**Description-**

1. **User Registration:**

Description: Allows new users to register and create an account on the platform.

Actors: User

Precondition: None

Postcondition: User account created successfully.

2. **User Login:**

Description: Enables users to log in to their account with valid credentials.

Actors: User

Precondition: User account exists

Postcondition: User successfully authenticated and logged in.

3. **Browse News Articles:**

Description: Allows users to browse through a list of available news articles.

Actors: User

Precondition: User logged in

Postcondition: User views the list of news articles.

4. **Search Articles:**

Description: Allows users to search for specific news articles based on keywords or topics.

Actors: User

Precondition: The user logged in

Postcondition: Search results displayed matching the user's query.

5**. View Article Details:**

Description: Allows users to view detailed information about a specific news article.

Actors: User

Precondition: The user selects a news article from the list

Postcondition: Detailed information about the selected article is displayed.

6. **Save Article:**

Description: Enables users to save interesting articles to their collections for later reference.

Actors: User

Precondition: The user logged in

Postcondition: Selected article saved to user's collection.

7. **Like/Dislike Article:**

Description: Allows users to express their preference for a news article by liking or disliking it.

Actors: User

Precondition: User logged in

Postcondition: User's preference recorded for the article.

8. **Personalized Recommendations:**

Description: Recommends news articles to users based on their preferences, past searches, and interaction history.

Actors: User

Precondition: User logged in

Postcondition: User receives personalized recommendations.

9. **Update Preferences:**

Description: Allows users to update their preferences for news topics or categories.

Actors: User

Precondition: User logged in

Postcondition: User preferences updated successfully.

10. **Logout:**

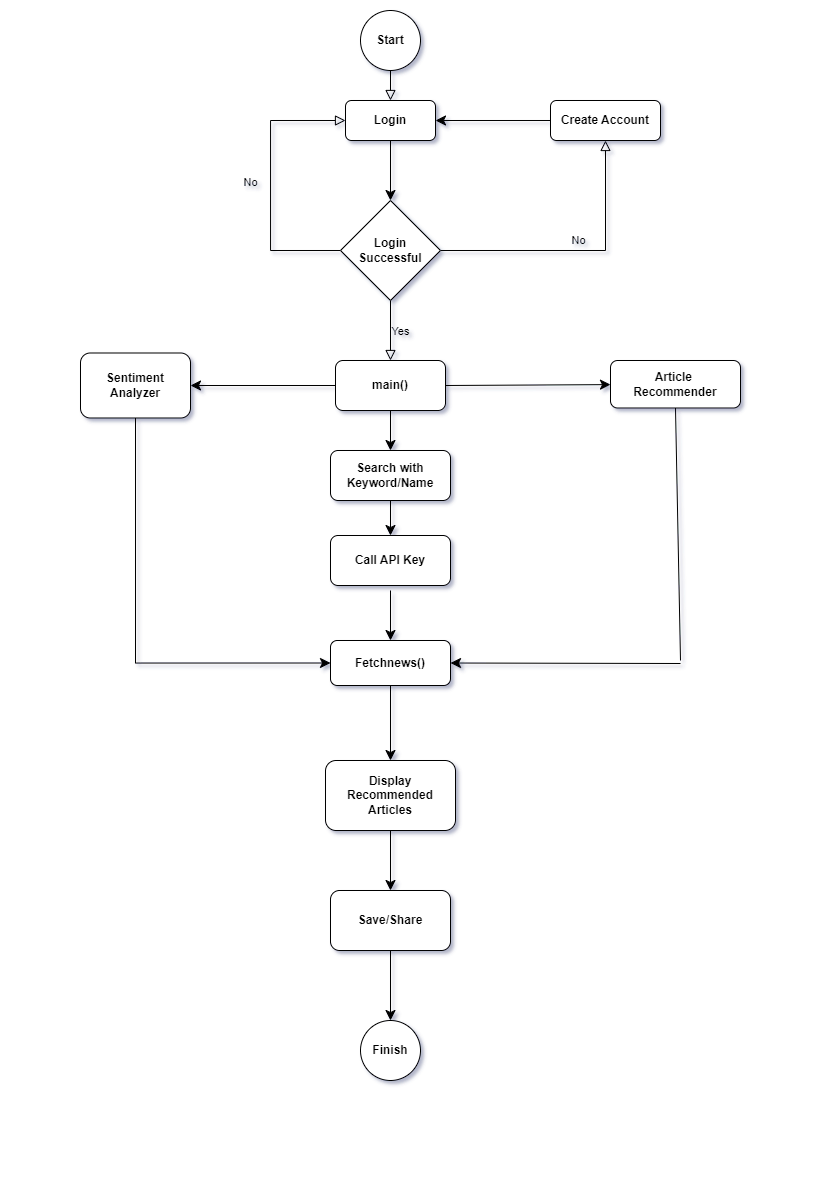
Description: Enables users to securely log out from their account.

Actors: User

Precondition: User logged in

Postcondition: User successfully logged out from the account.

***Activity Diagram-***

**

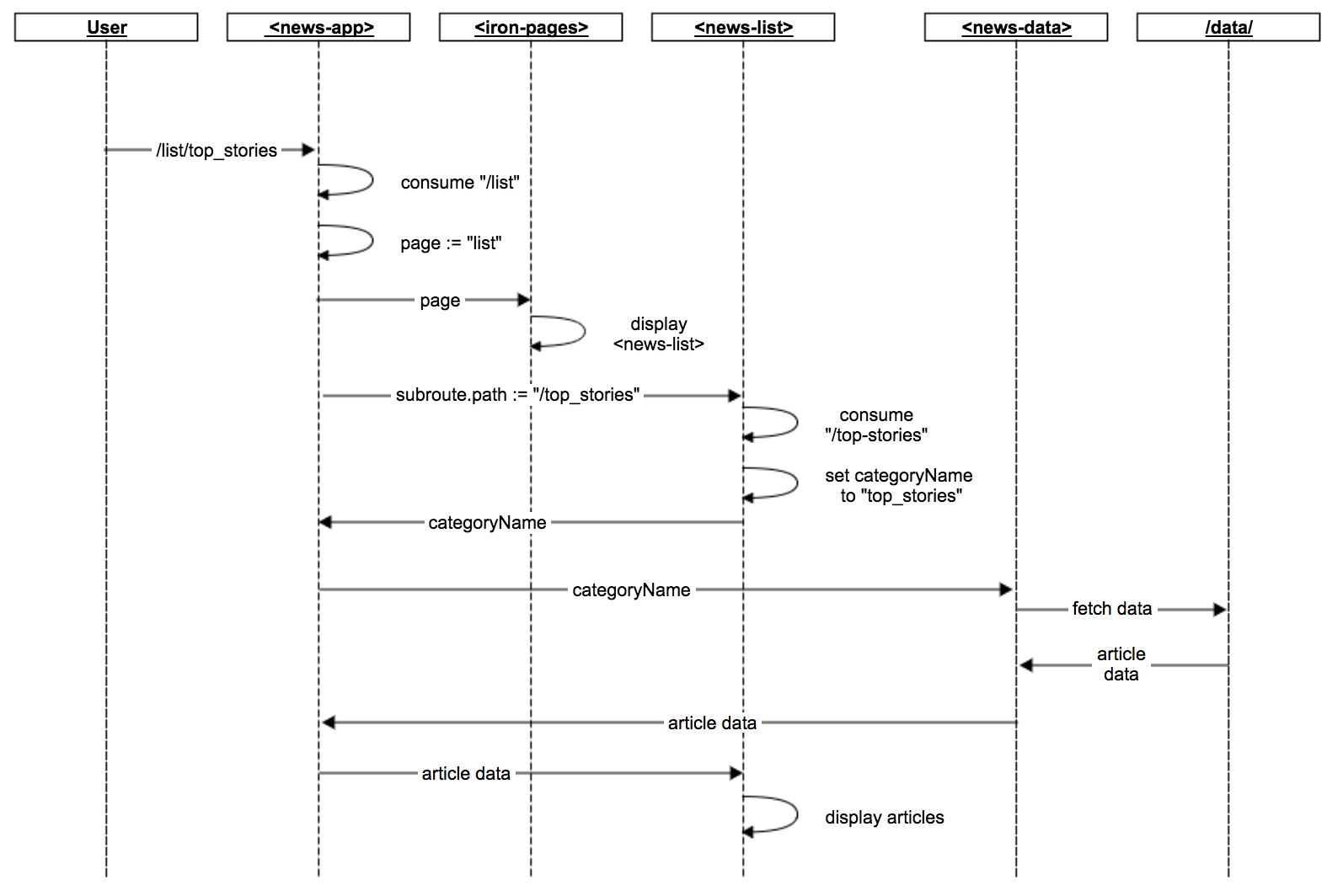
**Description-**

The activity diagram illustrates the process flow of user interaction with the personalized news aggregator system, from accessing news articles to receiving personalized recommendations based on user preferences and past interactions.

**Activities:**

1. **Access News Platform:**
   * **Start:** User accesses the personalized news aggregator platform.
   * **Action:** The user logs in to their account or registers as a new user if not already logged in.
2. **Browse News Articles:**
   * **Action:** User browses through the list of available news articles.
   * **Decision:** The user selects a specific article or performs a search for articles based on keywords or topics.
3. **View Article Details:**
   * **Action:** User views detailed information about the selected news article.
   * **Decision:** The user decides whether to read the full article, save it for later, or interact with it (like/dislike).
4. **Personalized Recommendations:**
   * **Action:** The system generates personalized recommendations for the user based on their preferences, past interactions, and browsing history.
   * **Decision:** The user receives and reviews the recommended articles.
5. **Update Preferences:**
   * **Action:** Users update their preferences for news topics or categories.
   * **Decision:** The user saves the updated preferences used to refine future recommendations.

Sequence Diagram-



Description-

This sequence diagram depicts the interaction between different components of a personalized news aggregator when a user requests a news feed.

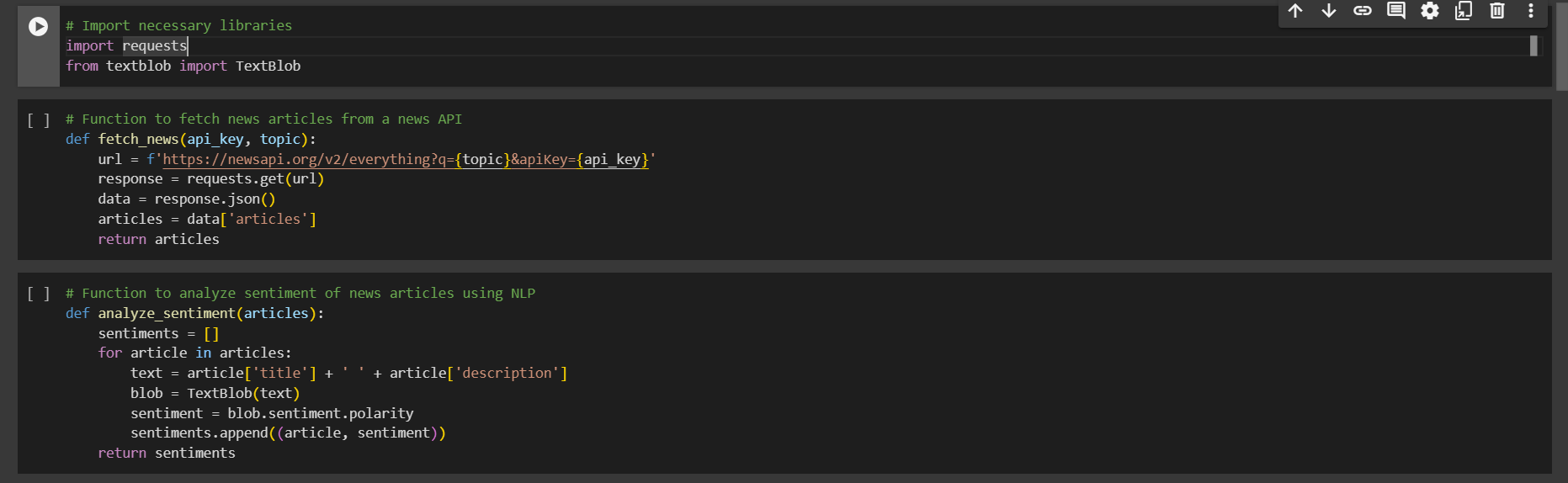
**Lifelines:**

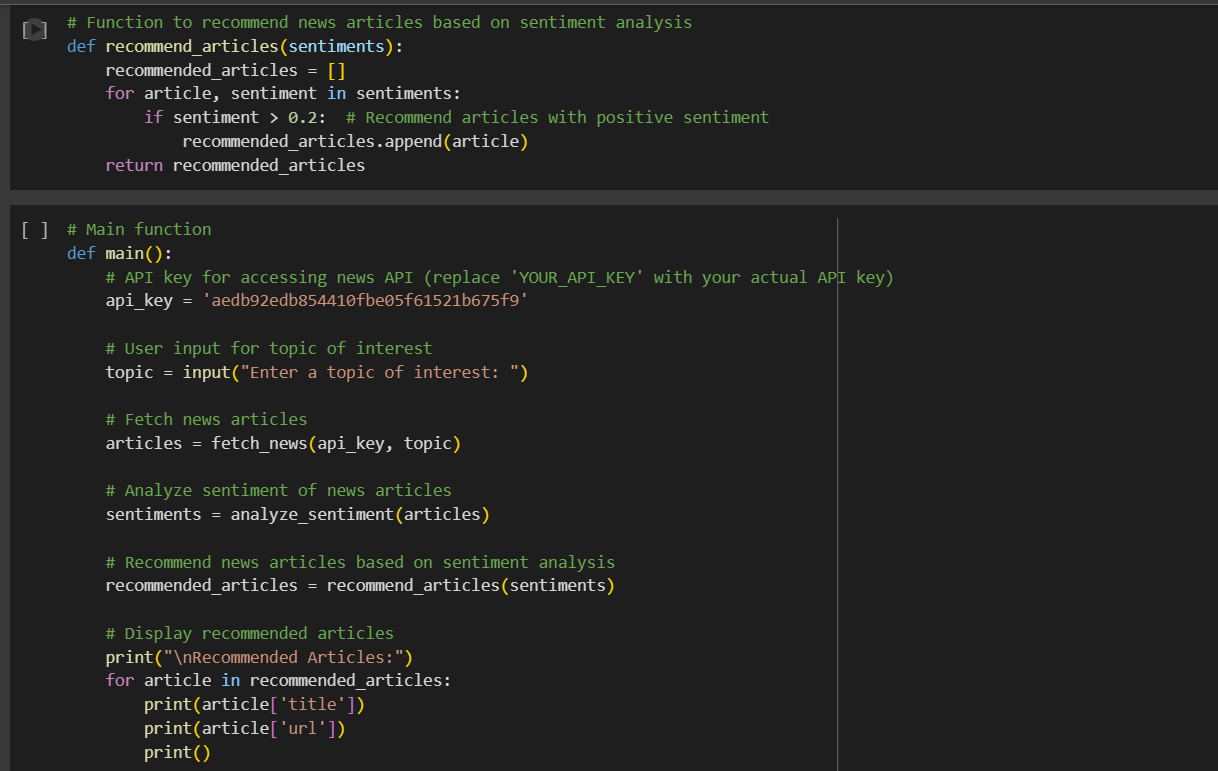
* **User:** Represents the user interacting with the application.
* **News Aggregator:** The core application responsible for fetching, personalizing, and displaying news articles.
* **User Profile:** Stores user information and preferences related to news topics, sources, and keywords.
* **News Source:** Represents external websites or feeds that provide news articles.
* **News Recommendation Engine:** Analyzes user data and news content to recommend relevant articles.

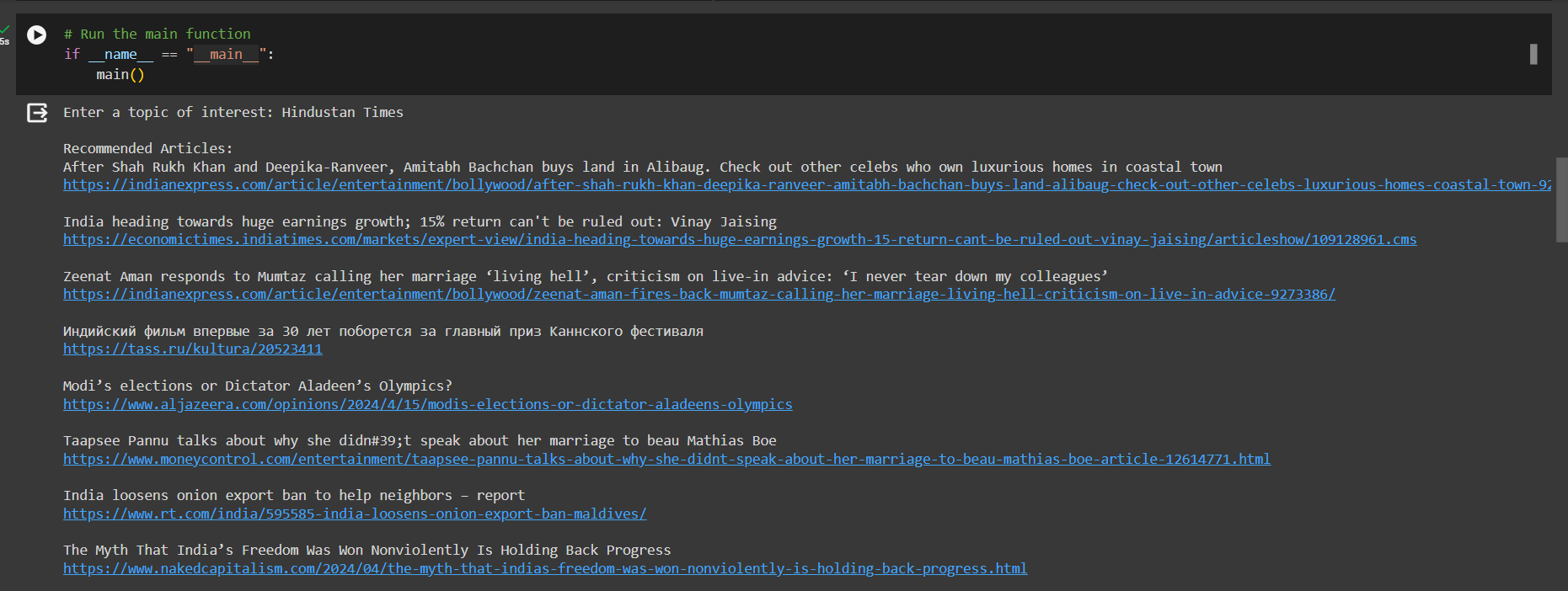
**Messages:**

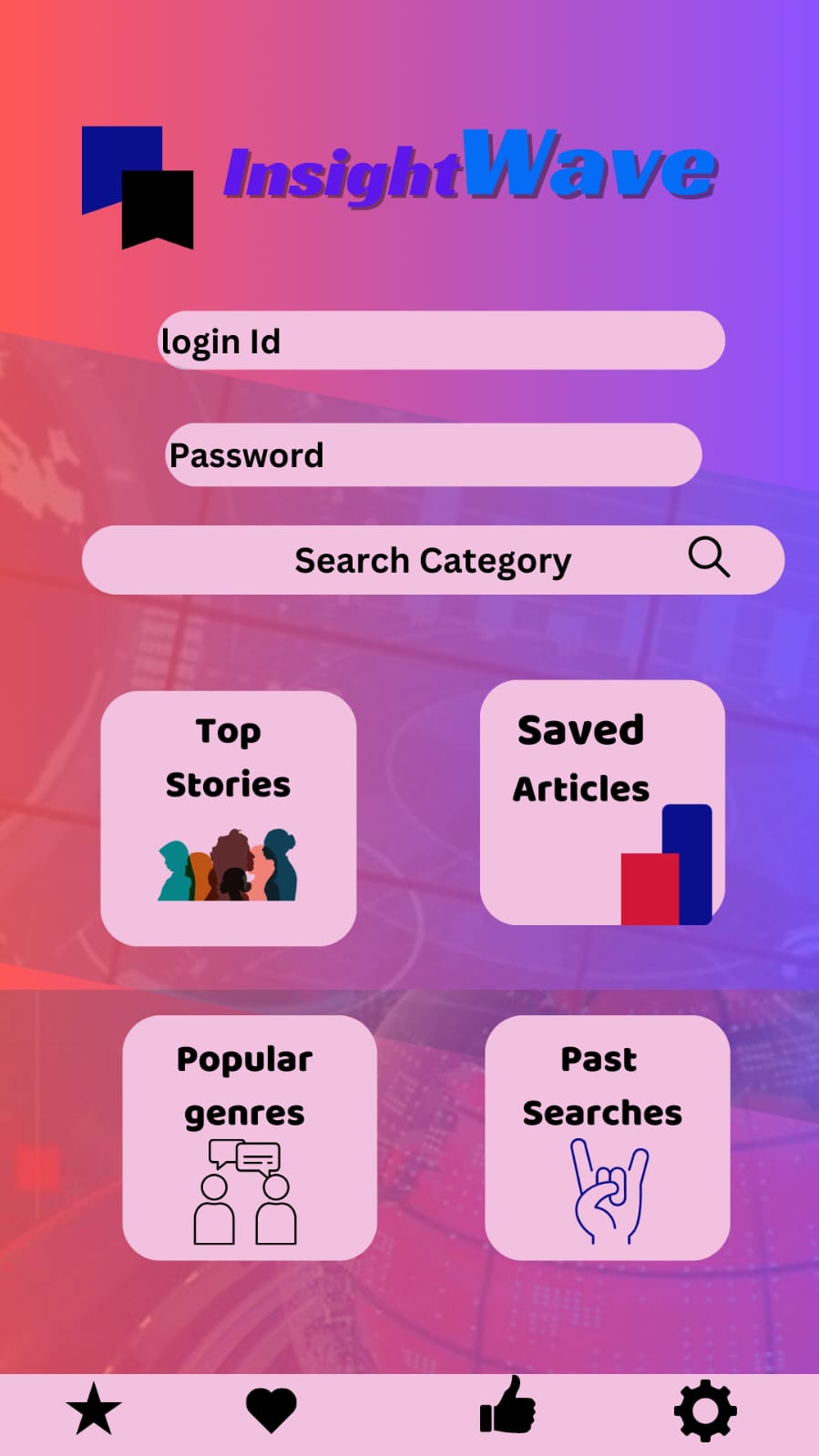
* **User -> News Aggregator:** "Request News Feed" - User requests a personalized news feed.
* **News Aggregator -> User Profile:** "Retrieve Preferences" - News Aggregator retrieves user preferences for personalization.
* **News Aggregator -> News Source(s):** "Fetch Articles" (multiple arrows) - News Aggregator fetches news articles from various sources based on user preferences (e.g., topics, keywords).
* **News Source(s) -> News Aggregator:** "Return Articles" (multiple arrows) - News sources send back-fetched articles.
* **News Aggregator -> News Recommendation Engine:** "Recommend Articles" - News Aggregator sends user preferences and fetches articles to the recommendation engine.
* **News Recommendation Engine -> News Aggregator:** "Recommend Articles" - The recommendation engine returns a list of recommended articles based on analysis.
* **News Aggregator -> User:** "Display News Feed" - News Aggregator displays the personalized news feed to the user (including recommended articles).

UI/UX Interface-

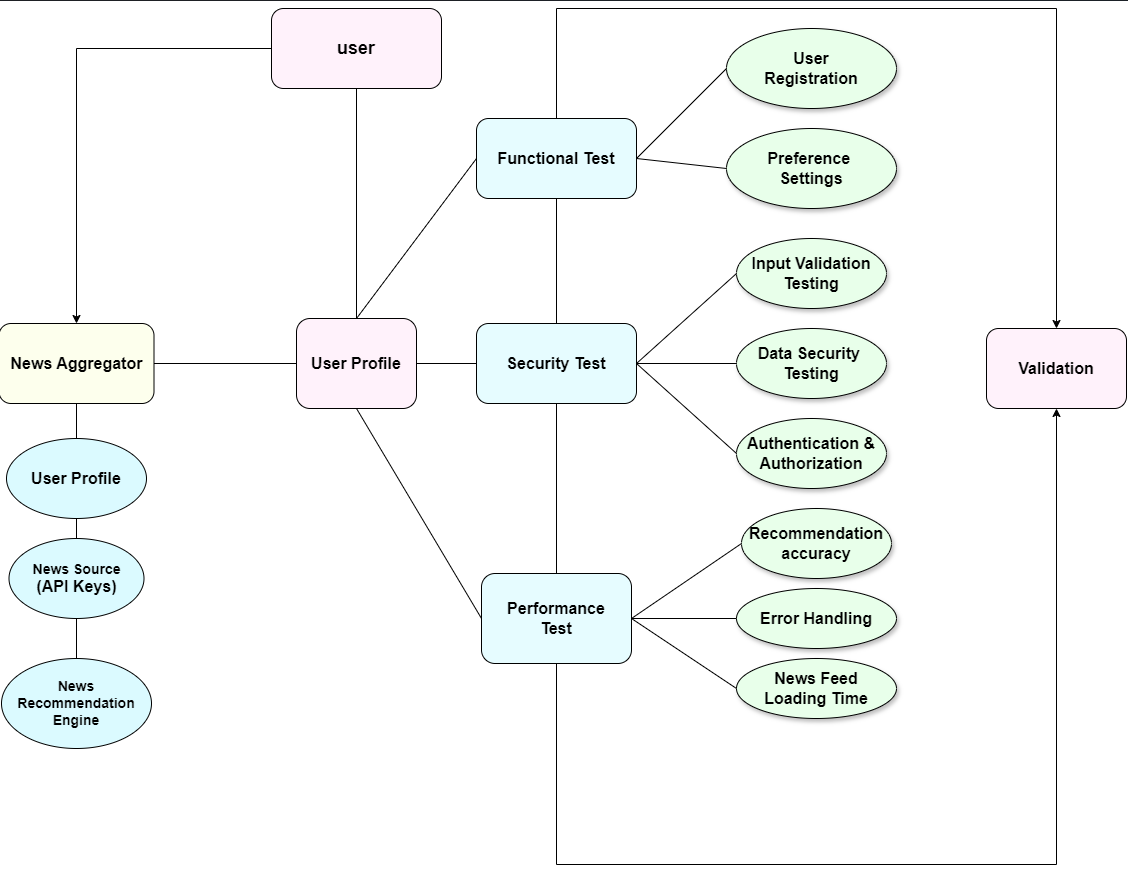








Testing-



**Description-**

The testing of the Personalized News Aggregator aims to ensure the reliability, effectiveness, and usability of the platform. It involves various types of testing to validate its core features and functionalities, as well as its performance under different conditions.

**Testing Types:**

**Functional Testing:**

* Functional testing focuses on validating the core features of the personalized news aggregator, including
* Testing the recommendation algorithm to ensure accurate and relevant article suggestions based on user preferences and past interactions.
* Verifying the functionality of browsing, selecting, and viewing news articles within the platform.
* Evaluating the user authentication and session management mechanisms to ensure secure access to personalized content.

**Performance Testing:**

* Performance testing assesses the efficiency and responsiveness of the platform, including:
* Testing the loading speed of news articles and recommendations to ensure optimal user experience.
* Assessing the platform's responsiveness under varying user loads and traffic conditions.
* Validating the system's ability to handle concurrent user interactions without degradation in performance.

**Security Testing**

* Security testing ensures the robustness of the platform against potential threats and vulnerabilities, including:
* Testing the encryption and secure transmission of user data to protect privacy and prevent unauthorized access.
* Assessing the platform's resistance to common security threats such as XSS (Cross-Site Scripting) and CSRF (Cross-Site Request Forgery).
* Verifying the effectiveness of authentication mechanisms in preventing unauthorized access to user accounts and data.

**Compatibility Testing:**

* Compatibility testing ensures the platform works seamlessly across different environments and devices, including:
* Testing the compatibility of the platform with various web browsers (Chrome, Firefox, Safari) and operating systems (Windows, macOS, Linux).
* Validating the responsiveness and usability of the platform on different screen sizes and resolutions, including desktops, laptops, tablets, and mobile devices.

**Test Scenarios:**

**Recommendation Algorithm:**

* Scenario 1: Verify the accuracy of article recommendations based on user preferences and past interactions.
* Scenario 2: Test the platform's response to changes in user preferences and browsing history.
* Scenario 3: Assess the diversity and relevance of recommended articles across different news categories.

**User Authentication:**

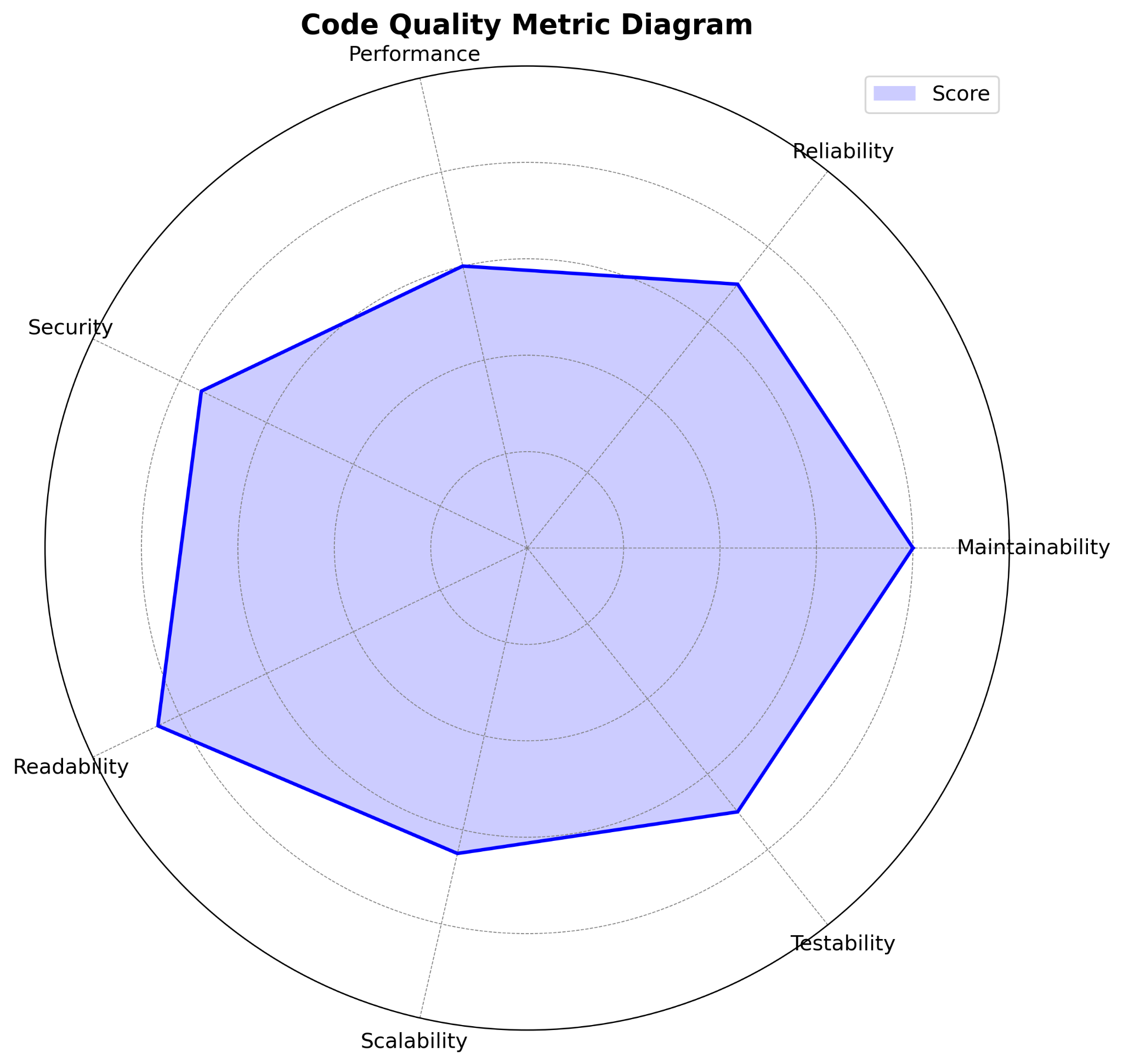
* Scenario 1: Test the authentication process for new users registering on the platform.
* Scenario 2: Verify the login process for existing users accessing their personalized news feed.
* Scenario 3: Assess the platform's response to invalid login attempts and password recovery requests.

**Article Browsing and Viewing:**

* Scenario 1: Test the functionality of browsing and selecting news articles from different categories.
* Scenario 2: Verify the display and formatting of articles for optimal readability and user experience.
* Scenario 3: Assess the platform's performance when loading and navigating between multiple articles simultaneously.

Quality Metrics:

Quality metrics in a personalized news aggregator refer to the criteria used to assess the effectiveness, performance, and user satisfaction of the platform. These metrics help evaluate various aspects of the aggregator's functionality, usability, reliability, and efficiency.



Description-

Quality Metrics for Personalized News Aggregator:

**1) Product Operation**:

* **Correctness:** While striving to meet user requirements, some features such as real-time chat support and integration of a chatbot for quick assistance were not implemented due to time constraints. However, core functionalities related to personalized news recommendations were successfully delivered.
* **Reliability:** Our software boasts a high level of reliability, leveraging Azure Cloud for hosting and ensuring scalability and uptime.
* **Usability**: The user interface of our platform is designed to provide a seamless experience, with personalized dashboards tailored to individual users' preferences. This approach minimizes friction and enhances user engagement.
* **Integrity:** To maintain data integrity and security, all stakeholders are required to undergo user authentication during registration, ensuring accountability for actions performed on the platform.
* **Efficiency**: Leveraging REACT JS for frontend development allows for efficient rendering of UI elements, minimizing processing requirements and optimizing performance.

**2) Product Revision:**

* **Maintainability and Flexibility**: The modular architecture of our software facilitates easy maintenance and upgrades. Each functionality is encapsulated within separate modules, allowing for independent modification and integration. Additionally, data flow between modules is managed using props, enhancing flexibility and adaptability.
* **Testability**: Rigorous testing procedures have been conducted to ensure the reliability and robustness of the software. Various testing methodologies, including unit testing, integration testing, and user acceptance testing, have been employed to validate the software's functionality and performance.

**3) Product Transition:**

* **Portability:** The software exhibits portability across multiple platforms, including Windows, Linux, Android, and iOS. Furthermore, the platform is designed to be responsive, ensuring optimal performance across various screen sizes and devices.
* **Reusability**: The UI elements and functionalities of the software are designed for reusability, allowing for easy integration into other systems. By simply modifying textual content and configurations, the software can be repurposed for different use cases, maximizing versatility and scalability.

**Precision**

Definition: The ratio of correctly identified user search categories to all data in the server. .

*Target:* Precision should be high, aiming for above 90%, to fetch relevant data.

**Recall (Sensitivity)**

*Definition:* The ratio of correctly identified user search categories to all actual data fetched from API keys.

*Formula: *

*Target:* Recall should be above 95% to ensure the most accurate recommendations are delivered..

**Response Time**

*Definition:* The time taken from user search to execution to deliver news accordingly.

*Unit:* Milliseconds (ms)

*Target:* Response time should be less than 500ms to minimize impact on user experience.

**Update Frequency**

Definition: How often the system updates its data provided by the user.

Unit: Updates per day

Target: Data should be kept updated to be recorded in the user’s past preferences.

**Compatibility Score**

*Definition:* A measure of how well the system works across different operating systems.

*Formula*:



*Target:* Compatibility Score should be 100% if the system supports all major operating systems (Android, iOS).

**CPU Utilization**

Definition: The percentage of CPU resources used during scanning or real-time protection.

Unit: Percentage (%)

Target: CPU utilization should be below 30% during normal operation to avoid system slowdowns.

**Memory Consumption**

Definition: The amount of RAM consumed by the news recommendation system.

Unit: Megabytes (MB)

Target: Memory consumption should be below 30MB to ensure efficient use of system resources.