Fall 2024-2025

CS 491 – Senior Design Project I

Analysis and Requirements Report

T2422 - Article Lens

Eray Yapağcı - 22103242

Mehmet Dedeler - 22003168

Alper Göçmen - 22002948

Mustafa Berkan Yıkılmaz - 22003325

Serra Çapraz - 22102314

Table of Contents

1. Introduction

In an era characterized by exponential growth in academic publications, researchers face significant challenges in navigating vast repositories of scholarly content to identify relevant and impactful studies. This deluge of information has rendered traditional research methods inefficient, often resulting in missed opportunities for innovation and collaboration. Recognizing this critical gap, the **Article-Lens** project proposes an AI-driven solution to streamline academic workflows and empower researchers with efficient literature discovery and exploration tools.

Article-Lens is an advanced research assistant designed to alleviate the burden of manual filtering and synthesis of academic content. It integrates cutting-edge technologies, including Large Language Models (LLMs), to deliver two primary features:

1. **Personalized Newsletter**: This component curates and delivers daily updates tailored to user-defined categories and topics, ensuring researchers remain informed about the latest field developments.
2. **Research Guide**: This feature provides structured learning paths, ranking references based on criteria such as relevance, citation count, and complexity, thereby facilitating a gradual transition from foundational to advanced knowledge.

By leveraging artificial intelligence for summarization, ranking, and personalization, Article-Lens aims to enhance the efficiency and accuracy of academic exploration. The project addresses the technical challenges of relevance filtering and data synthesis and emphasizes ethical considerations such as data privacy, transparency, and fairness in ranking algorithms.

This report outlines the analysis, requirements, and design considerations for Article-Lens, providing a comprehensive roadmap for its development and implementation. The project seeks to contribute meaningfully to the academic ecosystem through this initiative by offering a reliable, scalable, and user-centric research tool.

2. Current System

The current landscape of academic research tools offers limited functionality regarding personalized recommendations, structured learning paths, and efficient summarization of vast scholarly content. While platforms such as Google Scholar, Scholarcy, and ResearchRabbit provide valuable services, they must address modern researchers' comprehensive needs [1][2][3]. An analysis of these tools highlights their limitations as follows:

1. **Google Scholar**: A widely used platform for discovering academic literature, Google Scholar excels in search capabilities and citation metrics. However, it lacks features for personalized recommendations, structured newsletters, or advanced ranking mechanisms based on user preferences or paper impact [1].
2. **Scholarcy**: This tool condenses research papers into concise summaries, simplifying complex texts into digestible formats. Despite its utility in summarization, Scholarcy does not provide a mechanism for ranking papers, guiding users through structured learning paths, or offering customizable recommendation systems [2].
3. **ResearchRabbit**: As an AI-powered research tool, ResearchRabbit enables users to explore academic literature through interactive visualizations and curated collections. However, its focus on visual exploration and organization extends beyond structured learning or daily updates tailored to user-defined queries [3].

Limitations of the Current System

The existing tools present the following critical gaps:

- **Lack of Personalization**: Most systems do not allow users to tailor recommendations based on specific interests, scoring criteria, or learning preferences.
- **Absence of Structured Learning Paths**: There is no guidance for users to navigate academic topics from foundational to advanced levels systematically.
- **Limited Summarization Quality**: Current summarization tools lack flexibility, often failing to accommodate user-defined sections or maintaining high levels of accuracy and coherence.
- **Inefficient Relevance Filtering**: Existing platforms need to use sophisticated AI models to filter irrelevant content effectively.
- **Inadequate Integration**: Many tools operate as isolated solutions, needing interoperability with complementary platforms and databases.

These shortcomings underscore the need for a holistic, AI-powered system like Article-Lens, which aims to bridge these gaps by integrating advanced summarization, personalized ranking, and structured learning capabilities into a unified platform.

3. Proposed System

3.1 Overview

The **Article-Lens** system is envisioned as an innovative research assistant that leverages artificial intelligence to address the inefficiencies and challenges in academic literature discovery and analysis. By integrating state-of-the-art technologies such as Large Language Models (LLMs) and advanced ranking algorithms, the system aims to provide users with a comprehensive suite of tools to enhance their research workflows.

The proposed solution encompasses two primary functionalities:

1. **Personalized Newsletter**: This feature enables users to receive daily updates from platforms like arXiv and Google Scholar, tailored to their selected categories and queries [1][4]. Summaries are generated using LLMs, ensuring a concise and accurate representation of key findings, while the ranking engine prioritizes high-impact papers based on user-defined criteria.
2. **Research Guide**: This feature constructs structured learning paths by ranking references extracted from articles or topics. The ranking is guided by publication date, citation count, and complexity, facilitating a progressive transition from foundational to advanced knowledge in any field.

The architecture of Article-Lens is designed with scalability, reliability, and user-centricity in mind. It comprises three main components:

- **Data Ingestion & Preprocessing**: Responsible for collecting, cleaning, and preparing research data from arXiv and Google Scholar APIs [1][4].
- **Core Processing Module**: The heart of the system, this module handles relevance filtering, summarization, scoring, and ranking using AI-powered engines.
- **User Interaction Layer**: A responsive web interface that supports seamless navigation, personalized customization, and timely delivery of updates via email or notifications.

By bridging the gaps in the current research tools ecosystem, Article-Lens aspires to revolutionize how researchers engage with academic content, making the discovery process more efficient, structured, and impactful.

3.2 Functional Requirements

The functional requirements of the **Article-Lens** system define the core operations and processes that the software must perform to meet its objectives. These requirements ensure that the system provides the essential features and functionality for users to effectively discover, summarize, and rank academic content.

User Initiation

- Users must be able to select one or more categories from platforms such as arXiv (e.g., "Computation & Language") to define the scope of papers retrieved [4].

- Users should be able to input specific queries (e.g., "Prompt Injection") to refine their searches for relevant academic publications.

Daily Paper Retrieval

- The system must automatically retrieve new papers daily from user-selected categories on arXiv and Google Scholar [1][4].
- Retrieved papers must include metadata such as titles, abstracts, authors, and publication dates.

Relevance Filtering

- The system must analyze the abstracts of retrieved papers to determine relevance to the user's queries using a binary classification (Yes/No).
- Papers deemed irrelevant must be excluded from subsequent processes.

Summary Generation

- The system must generate summaries for each relevant paper, segmented into predefined sections such as "Findings" and "Framework."
- Users must be flexible to add or modify summary sections, like a "Disadvantages" section with proper description to enable custom summaries.
- Summaries must be tailored to user preferences using LLMs for accurate and concise generation.

Paper Ranking

- Papers must be scored and ranked using the following criteria:
    - **Author Scoring**: Based on metrics such as h-index and institutional reputation.
    - **Paper Scoring**: Evaluated for originality, impact, and relevance.
- Users must be able to adjust scoring weights and add custom classifications for personalized rankings.

Personalized Newsletter Delivery

- The system must aggregate the top-ranked papers into a daily newsletter.
- Users should have the option to receive the newsletter via email or access it through the platform.

Research Guide

- Users must be able to input a specific article or topic to generate a structured learning path.
- The system must extract references from external databases and rank them based on the following:
    - Publication date.
    - Citation count.
    - Complexity level (introductory or advanced).

- An ordered list of papers must be presented, starting from foundational to advanced works.

User Account Management

- Users must be able to create and manage accounts, including their preferences and saved queries.
- The system must securely store user data, such as search history and scoring criteria.

Notification System

- The system must notify users of new papers, updates in their selected categories, or changes to their research guide.
- Notifications must be customizable based on user preferences.

3.3 Non-Functional Requirements

The non-functional requirements of the **Article-Lens** system define the quality attributes the platform must meet to ensure usability, reliability, efficiency, and scalability. These requirements emphasize the operational aspects of delivering a robust and user-friendly experience.

Usability

- The platform must feature an intuitive, responsive user interface that facilitates easy navigation and operation.
- Users should be able to customize summary sections, scoring weights, and notifications without requiring technical expertise.
- The system must maintain compatibility across various devices, including desktops, tablets, and mobile devices, ensuring a seamless experience.
- Accessibility standards, such as WCAG (Web Content Accessibility Guidelines), must be adhered to for users with disabilities [5].

Reliability

- The system must ensure >99.6% uptime, providing consistent availability to users.
- Mechanisms must be implemented to prevent data loss and ensure the correctness and integrity of retrieved and processed data.
- Error handling and recovery systems should ensure smooth operation in case of unexpected failures.
- Frequent and automated data backups must be scheduled to protect against data loss.

Performance

- The platform must process and rank daily paper updates within five minutes of retrieval.

- AI summarization and ranking should be optimized for speed, ensuring a smooth user experience even with increasing data volume.
- Computational resources, especially for LLM operations, must be efficiently utilized to avoid bottlenecks.

Supportability

- The system architecture must support modularity, allowing for easy maintenance and integration of new features or updates.
- Comprehensive documentation must be provided for both developers and users, including user guides, FAQs, and troubleshooting resources.
- Logging and monitoring tools must be implemented to identify and resolve issues promptly.

Scalability

- The system must scale efficiently with increasing numbers of users and data volume, ensuring consistent performance.
- Cloud/server infrastructure should be utilized to allocate resources based on demand with consistent uptime.
- Databases must support high read/write operations without performance degradation.
- Load balancing strategies must be implemented to distribute workloads across servers evenly.

Security

- To protect user information, the system must adhere to data security standards such as ISO/IEC 27001 [6].
- Secure authentication mechanisms, such as multi-factor authentication, must be implemented to protect user accounts.
- User data must be encrypted during transmission and storage to ensure privacy and compliance with regulations like GDPR and KVKK [7][8].

By meeting these non-functional requirements, the **Article-Lens** platform will ensure a robust, scalable, and user-centric experience, accommodating the diverse needs of its user base.

3.4 Pseudo Requirements

Pseudo-requirements refer to constraints or conditions the system must comply with but are not directly tied to its functional or non-functional behavior. These requirements often arise due to environmental, technological, or legal factors and ensure that the system operates within specified boundaries.

Platform Compatibility

- The system must be developed as a web-based platform accessible through modern browsers such as Chrome, Firefox, Safari, and Edge.

- It should support a responsive design to ensure compatibility across different screen sizes and devices, including desktops, tablets, and smartphones.

Technology Stack

- The backend of the system must be implemented using Python Django, ensuring robust API development and seamless interaction with the front end [9][10][11][12].
- PostgreSQL must be used as the database management system to store metadata, user preferences, and summaries [13].
- The front end should be developed using React, providing a dynamic and interactive user experience [14].
- Integration with external APIs, such as arXiv and Google Scholar, must comply with their usage policies and rate limits [1][4].

Legal and Ethical Compliance

- The system must adhere to data privacy regulations, including GDPR (General Data Protection Regulation) and KVKK (Personal Data Protection Law in Turkey), ensuring secure handling of user information.
- Academic data must be processed in compliance with intellectual property laws, limiting access to open-access or fair-use content only.

Hosting and Infrastructure

- The platform should be hosted on a cloud service provider capable of providing access from an URL with consistent uptimes and scalability options remaining, such as AWS or Google Cloud [15][16].
- A continuous integration and deployment (CI/CD) pipeline must be implemented to streamline updates and maintenance [17].

Development and Testing

- Development must follow Agile methodologies, with iterative cycles for feature delivery and feedback integration [18].
- To ensure quality and reliability, the system must undergo rigorous testing, including unit testing, integration testing, and user acceptance testing (UAT).

User Feedback Integration

- Mechanisms must be implemented to gather user feedback regularly, enabling iterative improvements to the platform.
- Feedback loops should inform future development cycles, prioritizing enhancements based on user needs and pain points.

By adhering to these pseudo requirements, the **Article-Lens** system will align with technical, legal, and operational constraints while maintaining flexibility for future enhancements.

3.5. System Models

3.5.1. Scenarios

3.5.1.1 Access Personalized Newsletter

Basic Information

- **Use-case Name:** Access Personalized Newsletter
- **Actor:** User, ArxivAPI
- **Entry Condition:** The user is logged in and provides category/topic preferences
- **Exit Condition:** A personalized newsletter is delivered to the user

Primary Flow of Events

1. The user selects categories from Arxiv
2. The user provides specific topic query
3. The system fetches papers from selected categories via ArxivAPI
4. For each paper in the fetched list:
    - The system checks relevance to the user's query
    - The system generates customized section summaries
    - The system calculates author metrics and paper scores
    - The system ranks papers based on aggregate scores
5. The system compiles final newsletter
6. The system delivers newsletter to the user

Alternative Flow 1

1. The user selects categories from Arxiv
2. The system attempts to fetch papers
3. ArxivAPI connection fails
4. The system notifies user of temporary unavailability

Alternative Flow 2

1. System processes papers
2. No relevant papers were found for the query
3. The system suggests query modification to the user

3.5.1.2 Generate Research Guide

Basic Information

- **Use-case Name:** Generate Research Guide
- **Actor:** User, GrobidServer
- **Entry Condition:** User uploads article or provides topic
- **Exit Condition:** A structured reading path is presented

Primary Flow of Events

1. User uploads article or provides topic
2. The system validates document format
3. GrobidServer processes document
4. The system extracts references and DOIs
5. The system retrieves citation metrics
6. The system analyzes paper complexity
7. The system generates ordered reading path
8. The system presents path to the user

Alternative Flow 1

1. User uploads article
2. The system detects unsupported file format
3. The system requests different file format

Alternative Flow 2

1. GrobidServer processes document
2. Reference extraction fails
3. The system notifies the user of processing error

3.5.1.3 Process Paper Summary

Basic Information

- **Use-case Name:** Process Paper Summary
- **Actor:** User, LLM
- **Entry Condition:** Paper is selected for summarization
- **Exit Condition:** Customized summary is generated

Primary Flow of Events

1. The system receives paper for processing
2. The system validates paper content
3. For each custom section defined:
    - The system prompts LLM with section requirements
    - LLM generates section content
    - The system validates generated content
4. The system compiles all sections
5. The system presents final summary

Alternative Flow

1. The system attempts summary generation
2. LLM fails to generate coherent content
3. System retries with modified prompts
4. If a retry fails, the system notifies the user

3.5.1.4 Calculate Author Metrics

Basic Information

- **Use-case Name:** Calculate Author Metrics
- **Actor:** ScholarlyAPI
- **Entry Condition:** Author information is available
- **Exit Condition:** Complete author metrics are calculated

Primary Flow of Events

1. The system receives author details
2. The system queries ScholarlyAPI for metrics
3. The system retrieves h-index
4. The system gets citation counts
5. The system analyzes institutional affiliations
6. The system calculates aggregate author score

Alternative Flow

1. The system queries ScholarlyAPI
2. API returns incomplete data
3. The system uses available metrics
4. The system marks missing metrics in the final score

3.5.2. Use Case Model

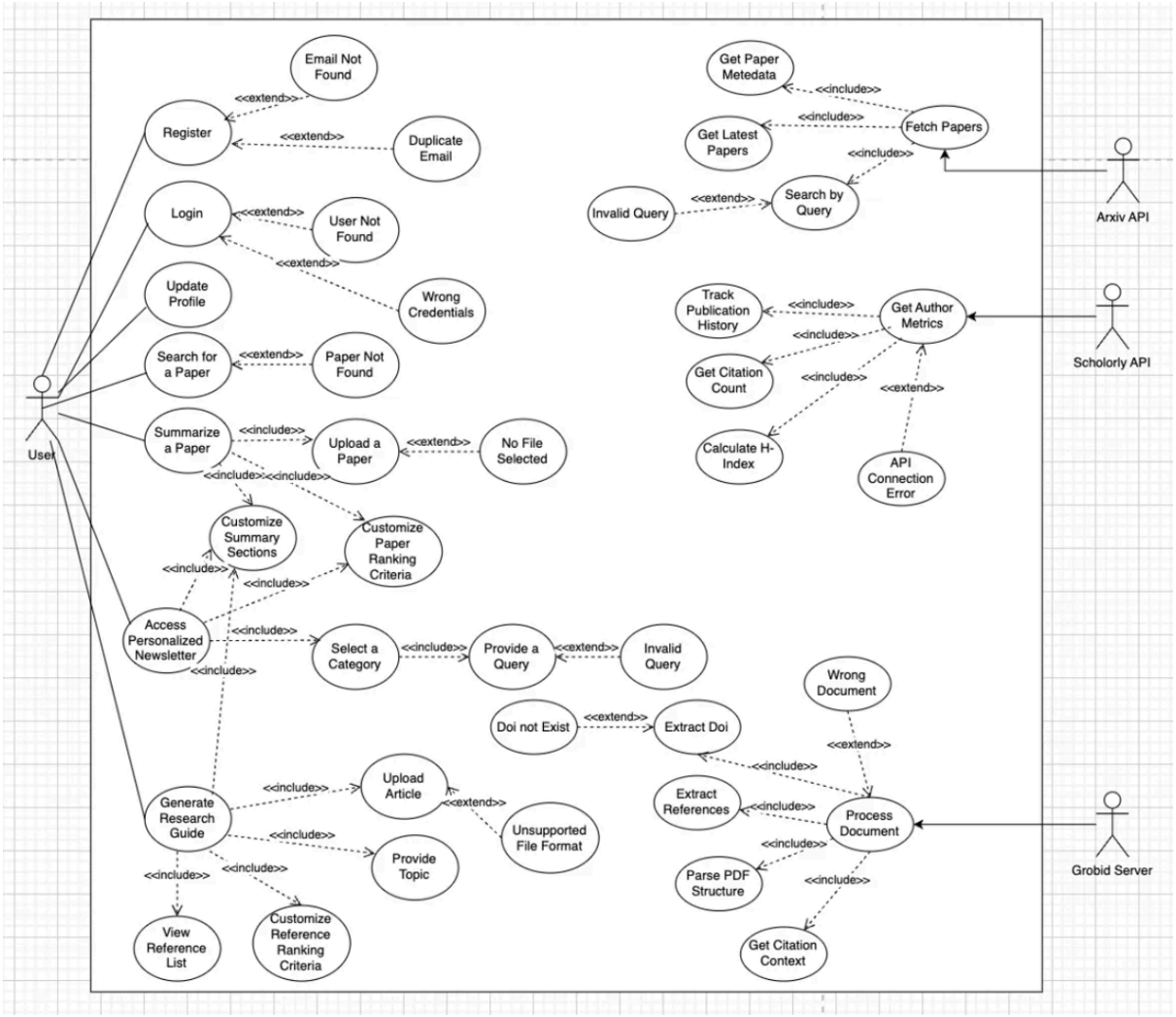


Figure 1. *Use Case Diagram*

3.5.3. Object and Class Model

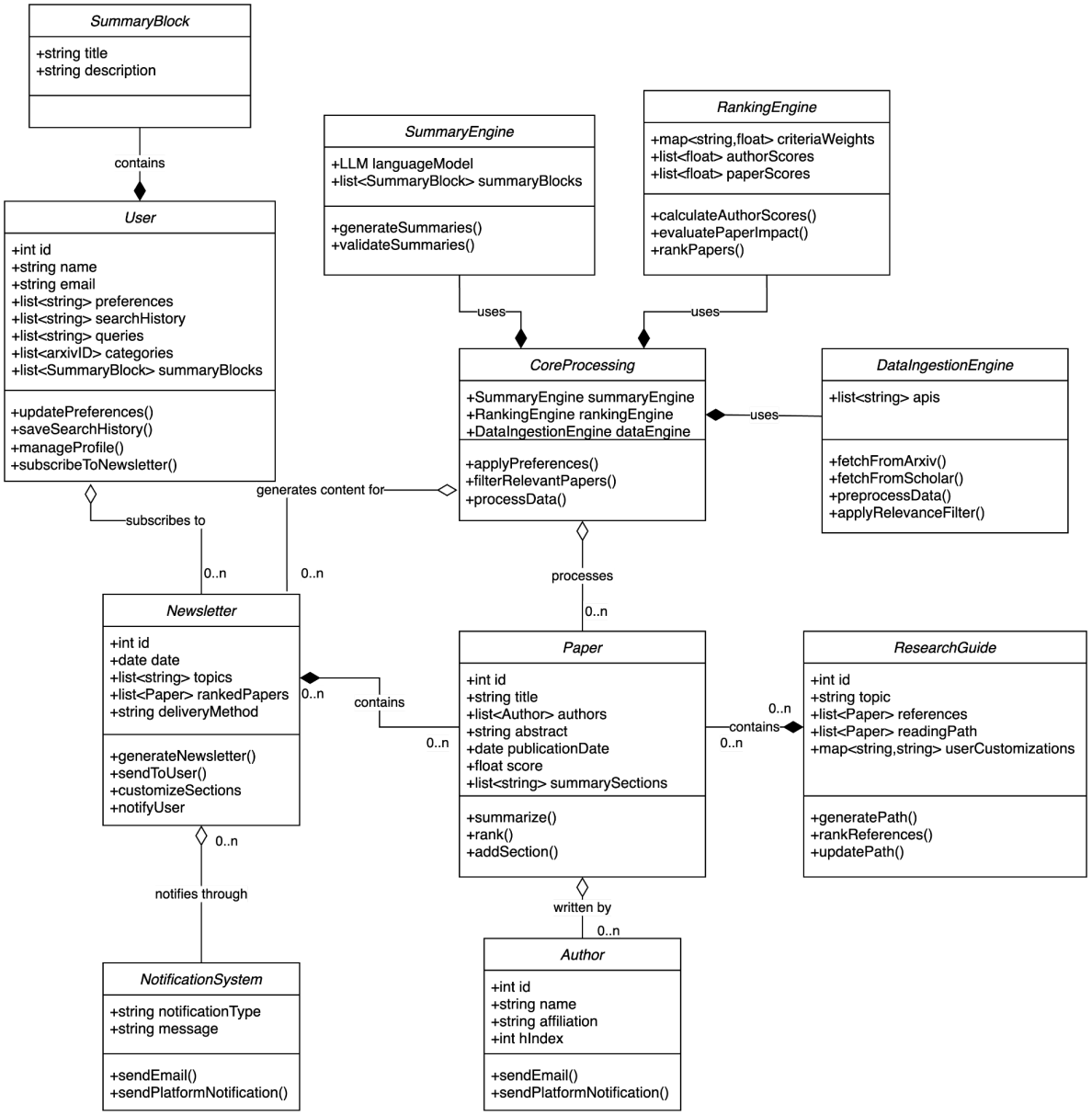


Figure 2. *Object and Class Diagram*

3.5.4. Dynamic Models

3.5.4.1 Newsletter Sequence Diagram

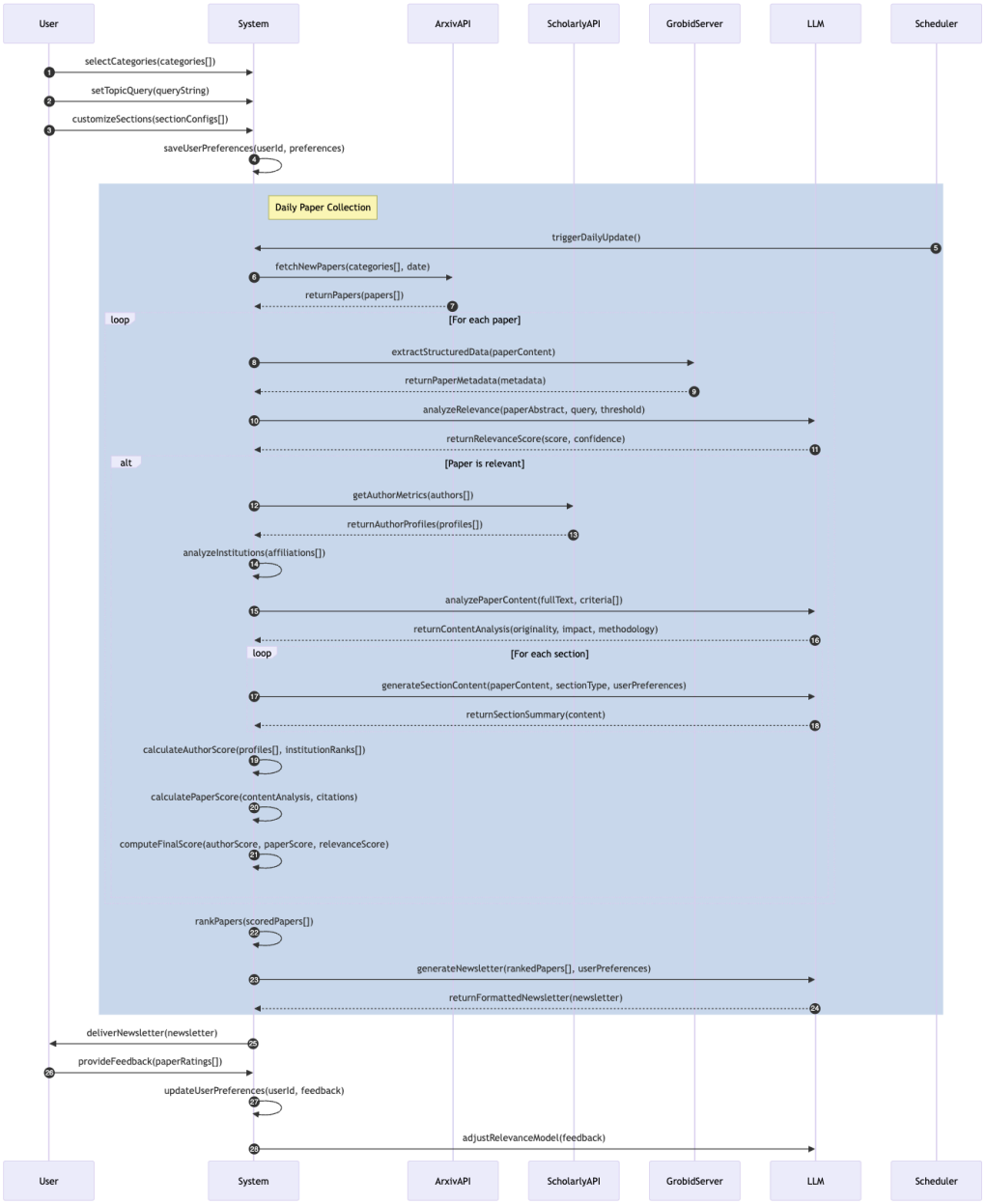


Figure 3. *Newsletter Sequence Diagram*

Newsletter Sequence Diagram: A daily automated system that processes research papers based on user preferences. It interfaces with Arxiv for paper collection, Grobid for paper processing, and Scholarly API for author metrics. The system filters relevant papers, analyzes content, generates customized LLM summaries, and delivers personalized newsletters. It includes a scoring mechanism that considers author impact, content quality, and user feedback for continuous improvement.

3.5.4.2 Research Guide Sequence Diagram

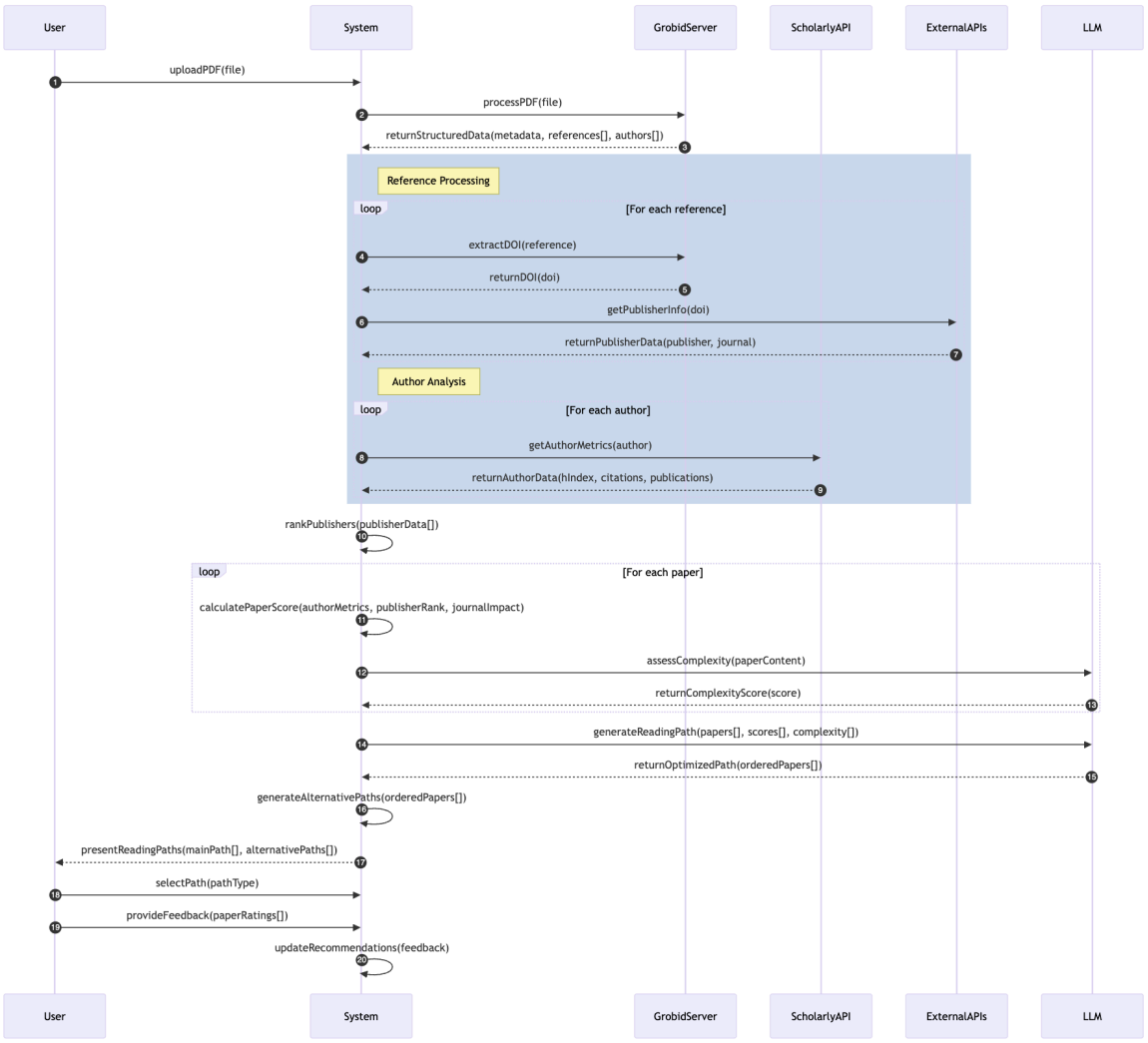


Figure 4. *Research Guide Sequence Diagram*

Research Guide Sequence Diagram: A system focusing on academic paper analysis through PDF processing. It uses the Grobid server to extract references and metadata, Scholarly API to analyze author impact, and implements publisher ranking. Creates multiple complexity-based reading paths through LLM analysis. Features user feedback integration for path optimization and allows alternative route selection based on user preferences. The primary purpose is to create structured learning paths from existing literature.

3.5.4.3 Newsletter Activity Diagram

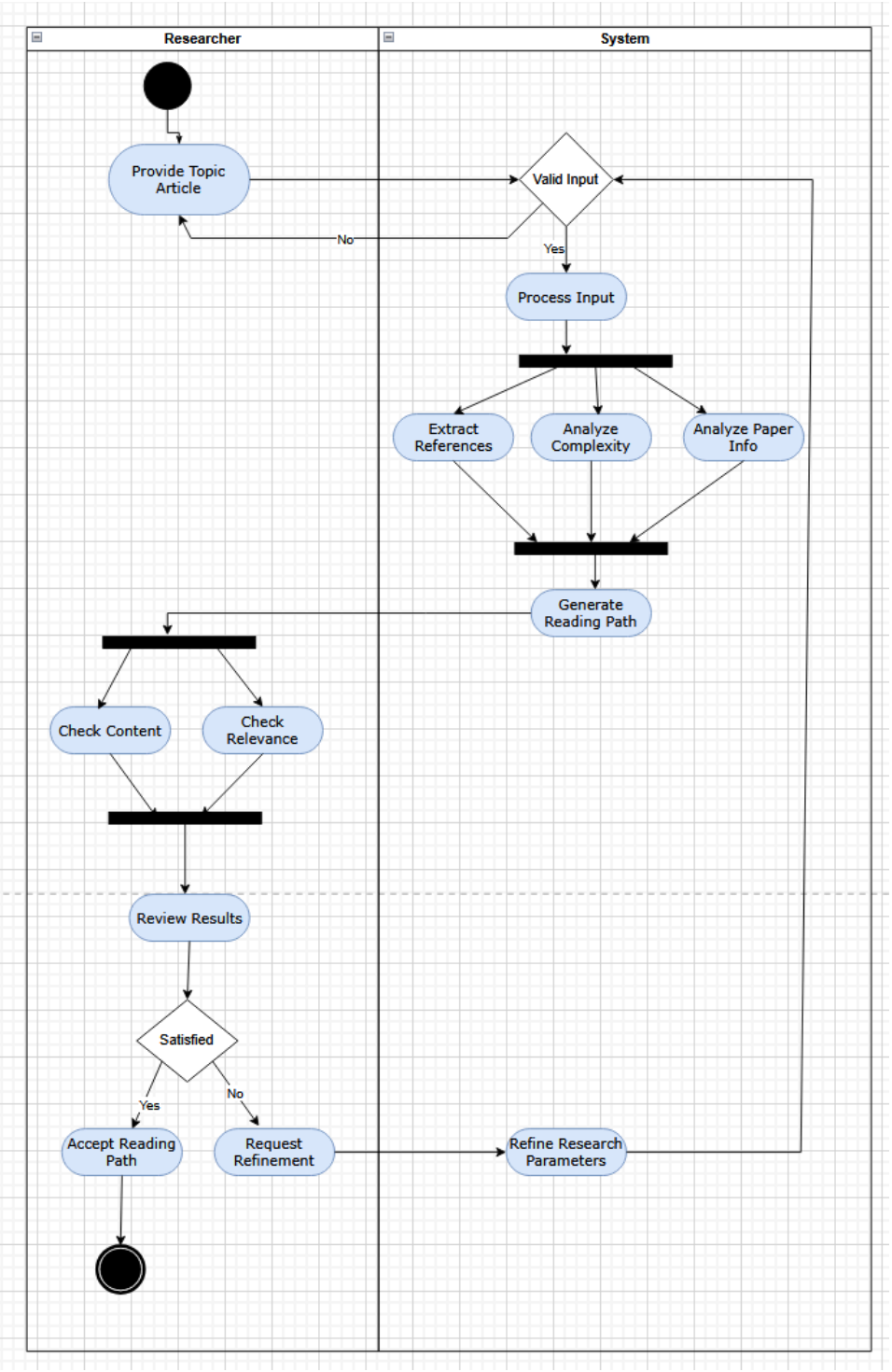


Figure 5. *Newsletter Activity Diagram*

The activity diagram represents a Reading Path Generator with two main actors: User and System. The User provides a research topic/article; then the System processes it through parallel operations (reference extraction and complexity analysis) to create a suggested reading path. The User reviews the path in parallel (checking content and relevance) and

either accepts it or requests refinements. This cycle continues until the User is satisfied with the reading path.

3.5.4.4 Research Guide Activity Diagram

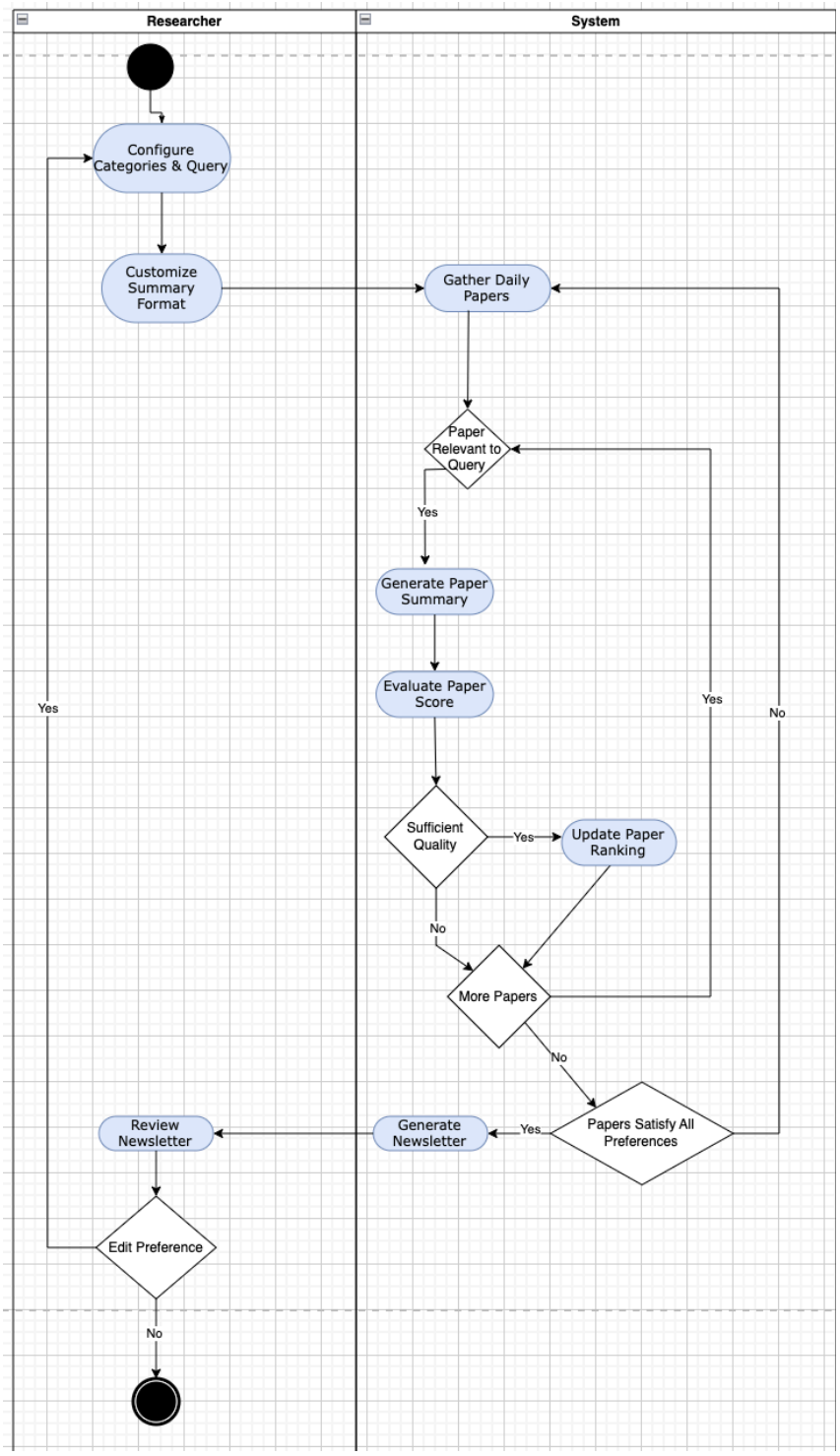


Figure 6. *Research Guide Activity Diagram*

The diagram illustrates an automated research paper filtering system where researchers configure search preferences and format settings while the system processes daily papers through multiple quality checks. The system evaluates each paper's relevance, generates summaries, and ranks them based on quality thresholds before compiling them into a

newsletter. If the researcher isn't satisfied with the results, they can modify their preferences, creating a continuous feedback loop for improved paper recommendations.

3.5.5. User Interface Mock-ups

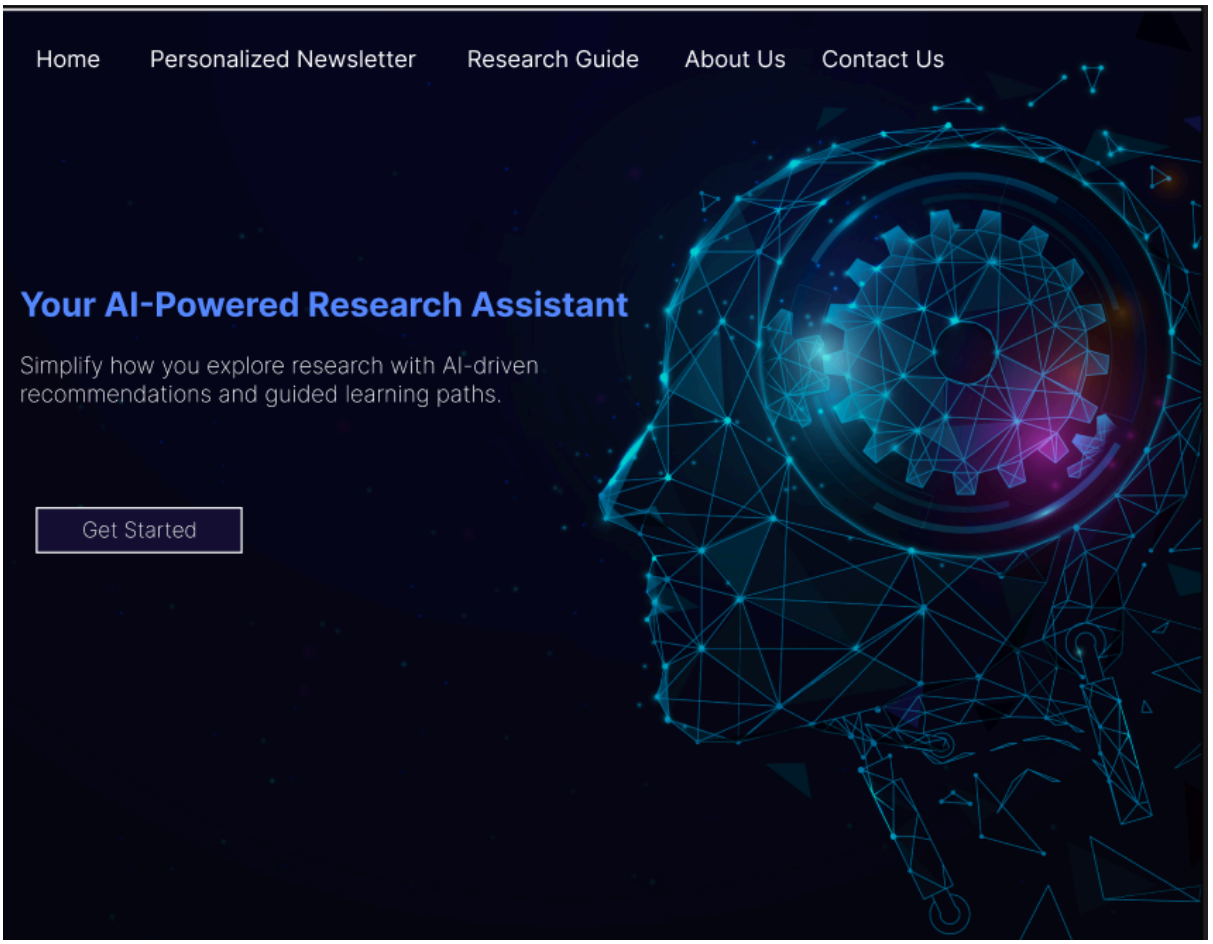
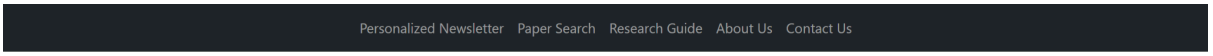


Figure 7. *Landing Page*



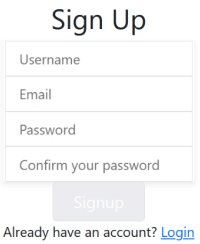


Figure 8. Sign-up Page with Basic User Registration Form

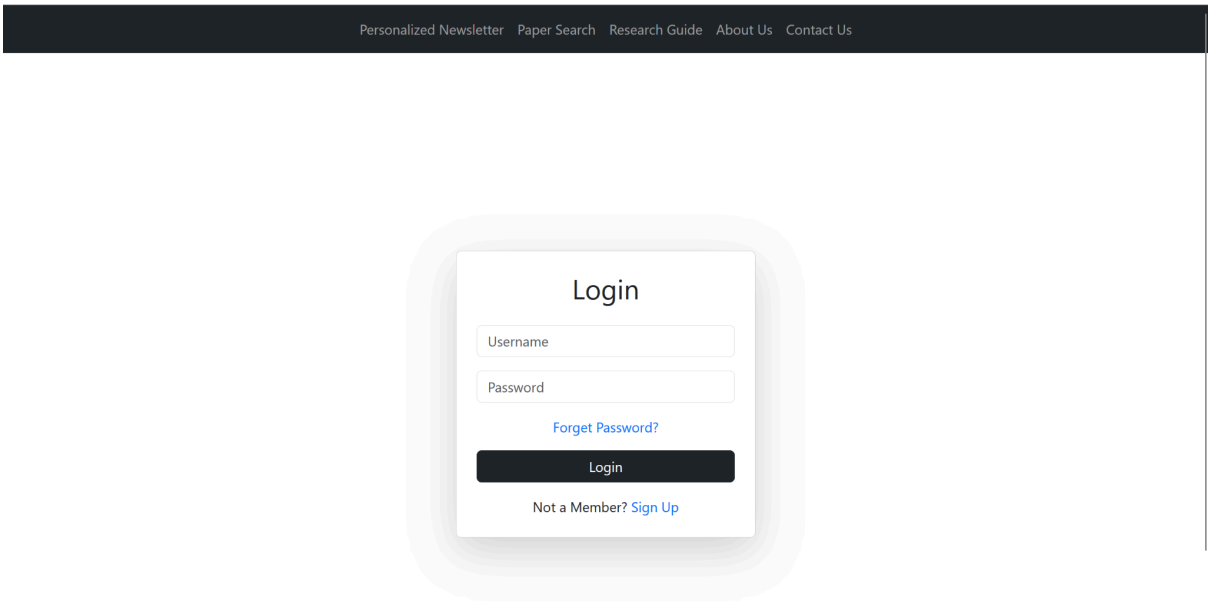


Figure 9. *Login Page*

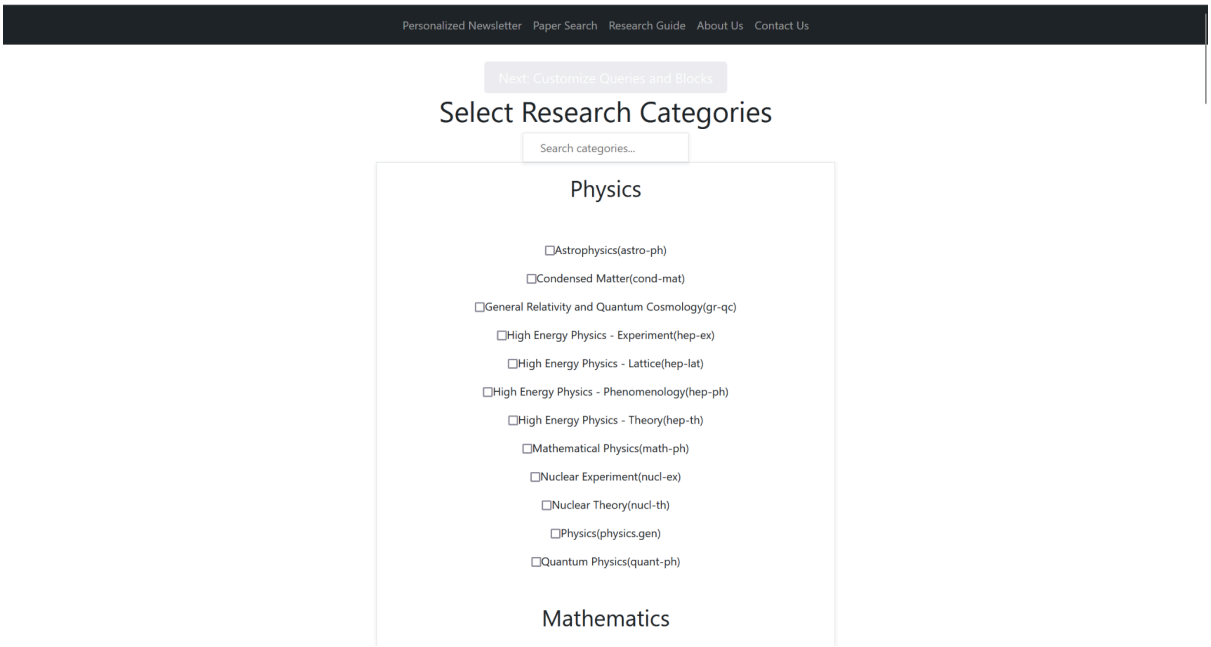


Figure 10. *Research Category Selection Page*
The page where contains the hierarchical list of academic fields (shown for Physics) where users can select their interests

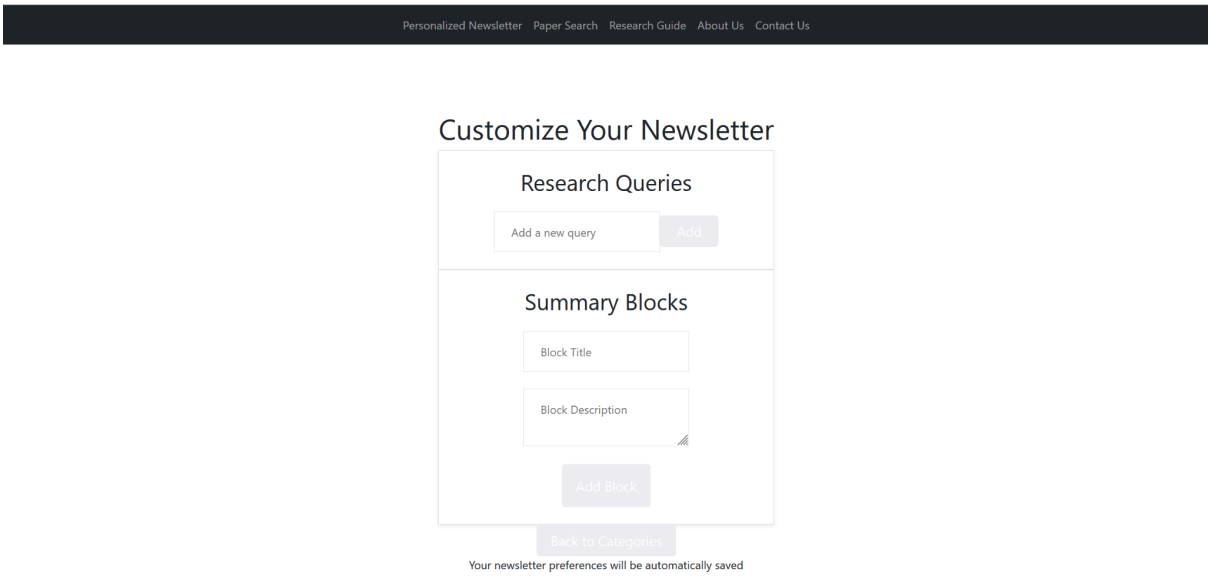


Figure 11. *Newsletter Customization Page*
The page where contains an interface for adding research queries and customizing summary blocks for personalized content

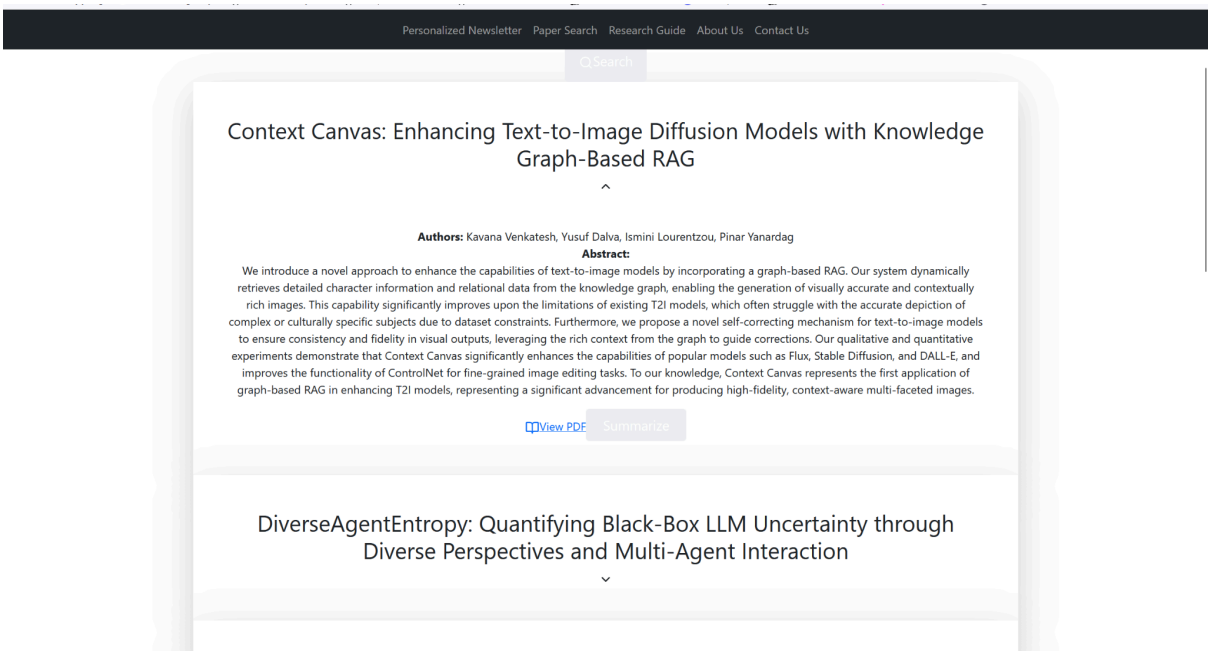


Figure 12. *Paper View Page*
The page that displays academic papers with title, authors, abstract, and expandable sections for summary and framework details

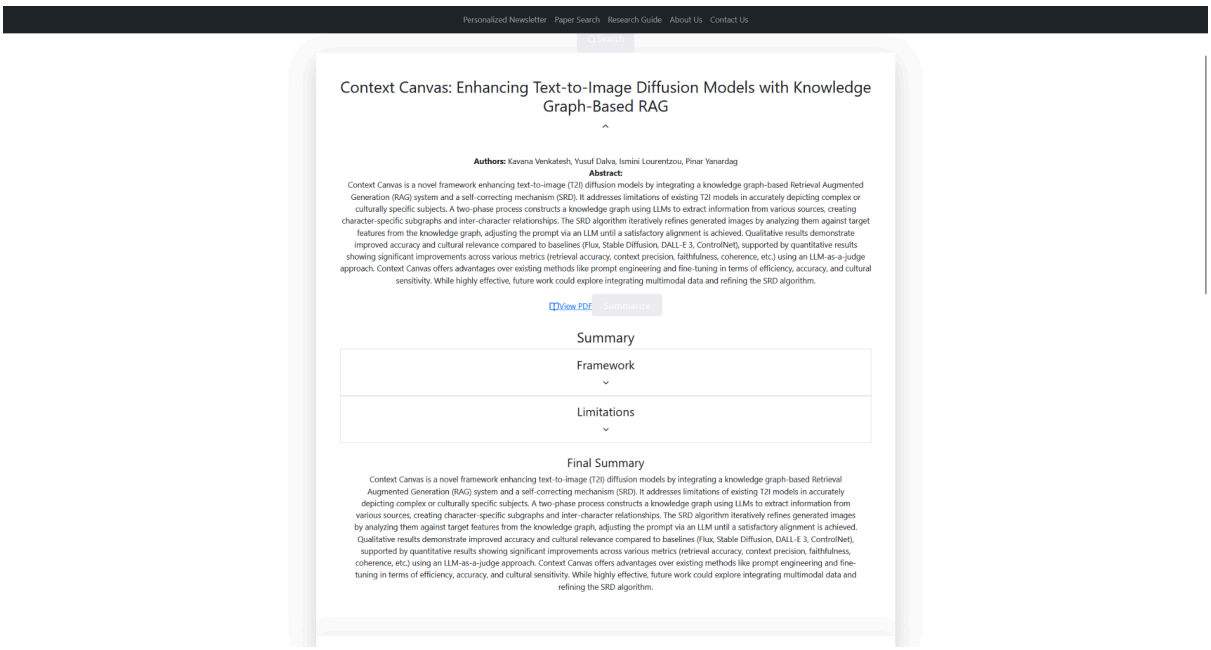


Figure 13. *Detailed paper view*

The page that shows comprehensive paper information with expandable sections for framework, limitations, and final summary

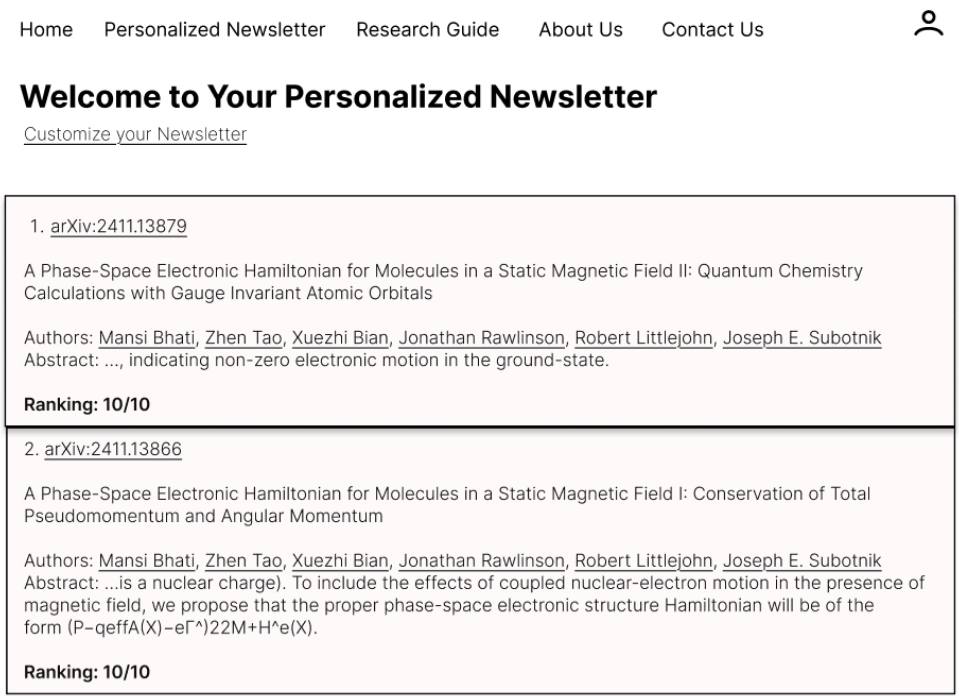


Figure 14. *Personalized Newsletter View*

The page that displays ranked research papers with key details such as title, authors, abstracts.

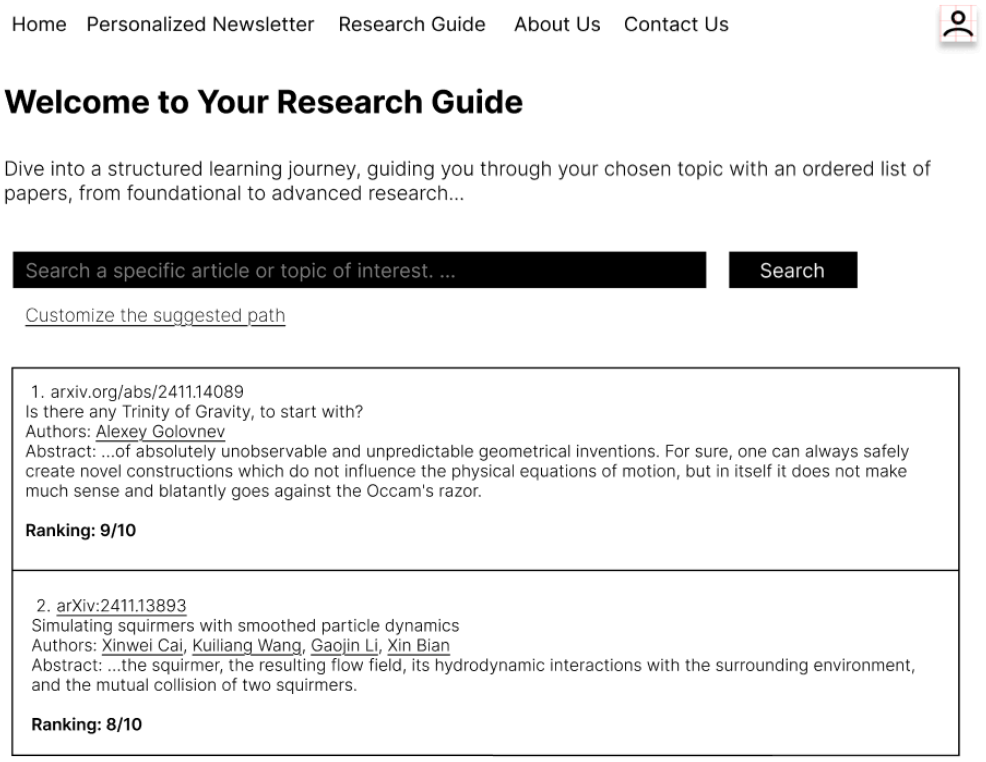


Figure 15. *Research Guide View*

The page that provides an ordered list of research papers based on the user's selected topic or article, offering a structured path from foundational to advanced studies, complete with rankings and abstracts.

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

The **Article-Lens** project incorporates various factors to ensure it meets the needs of stakeholders while addressing ethical, social, and global challenges. These considerations are critical for delivering a system that positively impacts users and society.

Public Health Considerations

The **Article-Lens** platform indirectly supports public health by enabling researchers to stay updated on the latest advancements in health-related fields such as epidemiology, medicine, and biomedical engineering. Providing a structured learning path and personalized summaries helps accelerate research that could lead to improved healthcare solutions. Accurate and timely dissemination of research findings through the platform can contribute to better-informed decisions in healthcare systems.

Public Safety Considerations

The platform prioritizes data security and privacy to ensure user safety. By adhering to data protection regulations like GDPR and KVKK, the system safeguards user preferences, search histories, and other sensitive information [7][8]. This focus on security not only protects users from potential data breaches but also builds trust in the platform. Furthermore, the accurate dissemination of research ensures that public safety decisions are based on reliable and well-vetted academic findings.

Public Welfare Considerations

By improving access to high-impact academic research, **Article-Lens** indirectly contributes to public welfare. The system enables researchers, educators, and professionals to access structured information, leading to advancements in education, technology, and healthcare. Over time, these improvements contribute to societal well-being by fostering innovation and supporting informed policymaking.

Global Considerations

The primary language of the platform is English, ensuring accessibility to a global audience. This enables researchers worldwide to benefit from the platform's features. Future plans include localization to support additional languages, making the system more inclusive. By adhering to global data protection and interoperability standards, **Article-Lens** can seamlessly integrate with academic systems across the world.

Cultural Considerations

The uniform nature of academic research minimizes the need for cultural customization in the platform. However, the system is designed to support diverse user groups by offering features that cater to varying levels of expertise. The platform promotes inclusivity by providing equal access to resources, regardless of cultural or geographic differences.

Social Considerations

The platform addresses social concerns by ensuring that user data is handled ethically and transparently. Privacy-focused features, such as anonymized data storage and encrypted communication, help mitigate ethical risks. Additionally, **Article-Lens** fosters collaboration among researchers, enabling users from different social and professional backgrounds to benefit from shared knowledge.

Environmental Considerations

By streamlining academic research workflows, the platform reduces the time and energy spent manually filtering and organizing research papers. In turn, this

Economic Considerations

The platform adopts a tiered monetization model to balance affordability and sustainability. While basic features are free to ensure accessibility for individual researchers, advanced functionalities are available through subscriptions. By integrating open-access APIs like arXiv, **Article-Lens** reduces reliance on expensive data sources, ensuring cost-efficiency for both developers and users [4].

4.1.1 Constraints

The **Article-Lens** project faces various constraints throughout its development and post-deployment stages. These constraints are categorized into implementation, economic, ethical, privacy, security, and usability constraints, as detailed below.

Implementation Constraints

- **AI Resource Dependency**: The project relies heavily on AI-based tools like LLMs for summarization and ranking, which require high-performance computing resources. These computational demands may increase operational complexity and costs.
- **API Limitations**: Platforms like arXiv and Google Scholar may impose API usage restrictions, such as rate limits or data access limitations, which could hinder seamless data integration.
- **Data Availability**: The effectiveness of relevance filtering and summarization depends on access to complete and accurate data. However, some articles or researchers may need more metadata or access, requiring the system to function efficiently in such scenarios.
- **System Complexity**: Providing customizable scoring and ranking criteria introduces system complexity, necessitating a robust yet user-friendly interface design.

Economic Constraints

- **Development Costs**: The project leverages free and open-source tools like Django and PostgreSQL to minimize initial costs. However, additional expenses may arise from premium API usage and cloud services [13][19].
- **Operational Costs**: The system's reliance on API calls to support LLM processing and server subscription data storage introduces recurring operational expenses.

- **Monetization**: A tiered subscription model is employed to sustain the platform financially. Basic features remain free for individual researchers, while advanced functionalities may require additional fees.
- **Scaling Costs**: As the user base grows, additional investments in computational infrastructure will be necessary to maintain system performance.

Ethical Constraints

- **Privacy Compliance**: Adherence to data privacy regulations, including GDPR and KVKK, is mandatory [7][8]. Robust mechanisms for anonymization and secure handling of user data must be implemented.
- **Algorithmic Bias**: Ranking algorithms may inadvertently favor certain authors, institutions, or articles. Mitigation strategies must be in place to ensure fairness and transparency in scoring.
- **Copyright Compliance**: Summarizing and using academic papers must respect copyright laws, restricting the system to open-access or fair-use content only.

Privacy and Security Constraints

- **Data Protection**: User data, including preferences and search histories, must be encrypted during storage and transmission to prevent unauthorized access.
- **Data Compliance**: Mechanisms for anonymization and secure storage of sensitive user information are required to meet legal and ethical standards.

User Experience and Usability Constraints

- **Target Audience**: Although the platform primarily targets researchers, the user interface must remain intuitive and accessible for non-experts or occasional users.
- **Flexibility and Accessibility**: The system must provide customizable features like scoring and ranking criteria without compromising ease of use or accessibility.

Addressing these constraints, the **Article-Lens** platform balances functionality, cost-effectiveness, ethical considerations, and user satisfaction.

4.1.2 Standards

Data Security Standards

- The system will adhere to **ISO/IEC 27001**, an international standard for managing information security, ensuring systematic identification and mitigation of security risks in handling user and research data [6].

Metadata Standards

- The **Dublin Core Metadata Standard** will be employed to organize and retrieve academic content effectively. This ensures consistency and efficiency in the storage and retrieval of metadata, including author details, publication dates, and keywords [20].

Usability Standards

- **ISO 9241**, the standard for ergonomics of human-system interaction, will guide the design of the user interface to ensure accessibility, user satisfaction, and ease of operation [21].

Interoperability Standards

- OpenURL and DOI standards will be incorporated to enable seamless integration with academic repositories and tools [22][23]. These standards facilitate direct linking to resources and consistent referencing, ensuring interoperability and access.

Article-Lens aims to deliver a secure, compliant, and user-friendly research platform by aligning with these standards and addressing the identified constraints.

4.2 Risks and Alternatives

Data Privacy and Security Risks

- **Risk**: Handling sensitive user preferences and search histories may expose the system to data breaches, resulting in privacy violations and potential legal consequences.
- **Mitigation**: Implement robust encryption methods for data storage and transmission. Enforce strict access controls and conduct regular security audits to ensure data privacy regulations (e.g., GDPR, KVKK) compliance.
- **Alternative**: Use only anonymized user data for personalization and maintain a minimal data retention policy to reduce the risk of misuse.

System Reliability Risks

- **Risk**: The dependency on external APIs such as arXiv and Google Scholar may result in disruptions if these services become temporarily unavailable or impose stricter rate limits [1][4].
- **Mitigation**: Implement caching mechanisms to store frequently accessed data and ensure partial functionality during API downtimes.
- **Alternative**: Explore partnerships with academic data providers to access datasets directly, reducing reliance on third-party APIs.

Performance and Scalability Risks

- **Risk**: The computational demands of LLM-based summarization and ranking may lead to slow response times or system overloads during peak usage.
- **Mitigation**: Optimize LLM operations by leveraging efficient API providers. Use load balancing to distribute requests evenly across servers.
- **Alternative**: Develop lightweight summarization models for basic functionality during high-load scenarios.

Ethical Risks

- **Risk**: The ranking algorithm may inadvertently introduce biases, favoring certain authors, institutions, or research topics, which could undermine user trust.

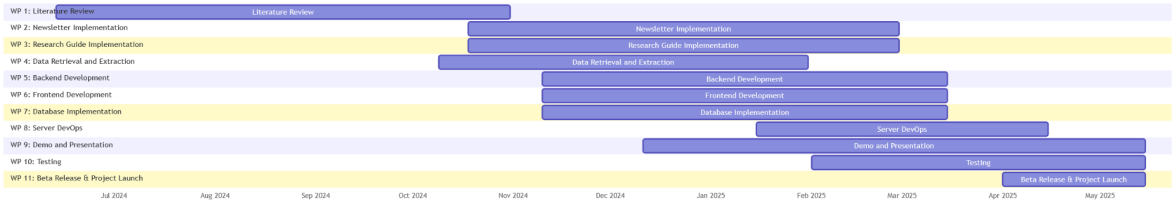
- **Mitigation**: Allow users to customize scoring criteria and provide transparency in ranking methodologies. Conduct periodic audits to identify and mitigate algorithmic biases.
- **Alternative**: Incorporate diverse evaluation metrics and consult with domain experts to improve the fairness of ranking systems.

Adoption and Usability Risks

- **Risk**: Resistance to adopting the platform due to unfamiliarity with its features or lack of integration with existing research workflows.
- **Mitigation**: Provide comprehensive user guides, tutorials, and responsive customer support to ensure a smooth onboarding experience.
- **Alternative**: Offer modular integration options that allow users to adopt specific features gradually, such as starting with the newsletter feature before using the research guide.

Risk Category	Likelihood	Impact	B Plan (Mitigation Strategy)
Data Privacy and Security	High	High	Encrypt data, enforce access controls, conduct regular audits, and use anonymized user data.
System Reliability	Medium	High	Implement caching mechanisms, use alternative data sources, and explore partnerships with data providers.
Performance and Scalability	Medium	High	Optimize LLM operations, optimize data collection, and consider mitigating cost-heavy operations.
Ethical Bias	Low	Medium	Allow user customization of scoring criteria, provide algorithm transparency and audit for biases.
Adoption and Usability	Medium	Medium	Provide user guides, tutorials, and customer support; allow modular adoption of features.

4.3 Project Plan



WP (Work Package)	WP Title	Leader	Members	Dates
WP 1	Literature Review	Eray Yapağcı	All Members	13 June 2024 - 31 Oct. 2024
WP 2	Newsletter Implementation	Eray Yapağcı	Mehmet Dedeler, Serra Çapraz	18 Oct. 2024 - 28 Feb. 2025
WP 3	Research Guide Implementation	Mehmet Dedeler	Eray Yapağcı, Alper Göçmen	18 Oct. 2024 - 28 Feb. 2025
WP 4	Data Retrieval and Extraction	Mustafa Berkan Yıkılmaz	Mehmet Dedeler	9 Oct. 2024 - 31 Jan. 2025
WP 5	Backend Development	Alper Göçmen	Mustafa Berkan Yıkılmaz, Serra Çapraz	10 Nov. 2024 - 15 Mar. 2025
WP 6	Frontend Development	Serra Çapraz	Eray Yapağcı	10 Nov. 2024 - 15 Mar. 2025
WP 7	Database Implementation	Alper Göçmen	Mustafa Berkan Yıkılmaz	10 Nov. 2024 - 15 Mar. 2025
WP 8	Server DevOps	Mustafa Berkan Yıkılmaz	Alper Göçmen, Mehmet Dedeler	15 Jan. 2025 - 15 Apr. 2025
WP 9	Demo and Presentation	Eray Yapağcı	All Members	11 Dec. 2024 - 15 May 2025
WP 10	Testing	Serra Çapraz	All Members	01 Feb. 2025 - 15 May 2025
WP 11	Beta Release & Project Launch	Mehmet Dedeler	All Members	01 Apr. 2025 - 15 May 2025

WP 1: ***Literature Review***			
Start date: ***13 June, 2024*** **End date:** ***31 October, 2024***			
Leader:	**Eray Yapağcı**	**Members involved:**	**All members**
Objectives: ***The goal is to review existing literature to understand current academic tools, identify gaps, and define state-of-the-art AI applications in academic research. This will help establish the foundational knowledge needed for the development of Article-Lens.***			
Tasks: **Task 1.1 Review Current Tools: Analyze existing academic tools to assess their functionalities and limitations.** **Task 1.2 Study AI Implementations: Study how AI is used in academic research tools and summarize findings.**			
Deliverables **D1.1: Comprehensive literature review report.** **D1.2: Analysis of AI applications in existing academic tools.**			

WP 2: ***Newsletter Implementation***			
Start date: ***18 October, 2024*** **End date:** ***28 February, 2025***			
Leader:	**Eray Yapağcı**	**Members involved:**	**Mehmet Dedeler, Serra Çapraz**
Objectives: ***Develop the personalized newsletter feature that filters, summarizes, and ranks new publications based on user-defined categories and queries, ensuring users receive timely and relevant academic updates.***			
Tasks: ***Task 2.1 Newsletter System Design: Design the architecture for the newsletter system, including the integration with the summarization engine.*** ***Task 2.2 Implementation and Testing: Implement and test the newsletter delivery system to ensure reliability and accuracy.*** ***Task 2.3 User Customization Features: Develop features that allow users to customize the frequency and content of their newsletters.***			
Deliverables ***D2.1: Functional newsletter system.*** ***D2.2: Test reports and system documentation***			

WP 3: *Research Guide Implementation*			
Start date: *18 October, 2024* **End date:** *28 February, 2025*			
Leader:	**Mehmet Dedeler**	**Members involved:**	**Eray Yapağcı, Alper Göçmen**
Objectives: ***Create a research guide feature that provides structured learning paths, ranking references based on relevance, complexity, and citation count, to aid users in transitioning from foundational to advanced knowledge levels efficiently.***			
Tasks: ***Task 3.1 Design Learning Path Algorithm: Develop algorithms to create and rank learning paths.*** ***Task 3.2 System Integration and Testing: Integrate the research guide with the core system and conduct comprehensive testing.***			
Deliverables ***D3.1: Research guide feature implementation.*** ***D3.2: Integration and test reports.***			

WP 4: *Data Retrieval and Extraction*			
Start date: *9 October, 2024* **End date:** *31 January, 2025*			
Leader:	**Mustafa Berkan Yıkılmaz**	**Members involved:**	**Mehmet Dedeler**
Objectives: ***Develop a module to handle the retrieval and extraction of data from arXiv and Google Scholar, ensuring the data is clean and structured for processing.***			
Tasks: ***Task 4.1 Develop Data Retrieval System: Build the infrastructure to fetch and extract data from specified APIs.*** ***Task 4.2 Data Cleaning Process: Implement data cleaning mechanisms to prepare raw data for further processing.***			
Deliverables ***D4.1: Data retrieval module.*** ***D4.2: Data preprocessing toolkit.***			

WP 5: *Backend Development*			
Start date: *10 November, 2024* **End date:** *15 March, 2025*			
Leader:	**Alper Göçmen**	**Members involved:**	**Mustafa Berkan Yıkılmaz, Serra Çapraz**
Objectives: ***Develop the backend infrastructure necessary to support the functionalities of Article-Lens, including data management, process orchestration, and API integration. This backend will serve as the backbone of the application, handling data transactions, authentication, and server-side logic.***			
Tasks: **Task 5.1 API Development: Create RESTful APIs to handle requests and serve data to the front end.** **Task 5.2 Backend Logic Implementation: Implement the core logic that powers the application's features, such as relevance filtering, summarization, and personalized ranking.** **Task 5.3 Security Implementation: Implement security measures to protect data and ensure privacy compliance.** **Task 5.4 Integration with External Services: Integrate backend services with external APIs and data sources for enhanced functionality.**			
Deliverables **D5.1: Complete backend system capable of handling specified functionalities.** **D5.2: API documentation and backend system guidelines.** **D5.3: Security audit report.** **D5.4: Integration success report for external services.**			

WP 6: ***Frontend Development***			
Start date: ***10 November, 2024*** **End date:** ***15 March, 2025***			
Leader:	**Serra Çapraz**	**Members involved:**	**Eray Yapağcı**
Objectives: ***Design and implement the user interface and experience, ensuring it is intuitive, responsive, and accessible across various devices and platforms. This work package aims to create a dynamic and visually appealing front end that facilitates easy interaction with the system's features.***			
Tasks: **Task 6.1 UI/UX Design: Design the user interface and experience based on best practices and user feedback.** **Task 6.2 Frontend Programming: Develop the frontend application using React to ensure dynamic interaction.** **Task 6.3 Accessibility Features: Implement accessibility standards to ensure the platform is usable by all users, including those with disabilities.**			
Deliverables **D6.1: User interface and experience design.** **D6.2: Functional frontend application.** **D6.3: Accessibility compliance report.**			

WP 7: ***Database Implementation***			
Start date: ***10 November, 2024*** **End date:** ***15 March, 2025***			
Leader:	**Alper Göçmen**	**Members involved:**	**Mustafa Berkan Yıkılmaz**
Objectives: ***Set up a robust database system to store and manage user data, preferences, and system-generated content such as summaries and newsletters. The database will be optimized for high performance and scalability to support a growing user base.***			
Tasks: ***Task 7.1 Database Design: Design a database schema that supports efficient data retrieval and storage.*** ***Task 7.2 Database Setup and Integration: Implement the database using PostgreSQL and integrate it with the backend.*** ***Task 7.3 Data Integrity and Backup Systems: Establish systems to ensure data integrity and implement regular backup procedures.***			
Deliverables ***D7.1: Database schema and design documentation.*** ***D7.2: Implemented and functional database system.*** ***D7.3: Data integrity checks and backup protocols.***			

WP 8: *Server DevOps*			
Start date: *15 January, 2024* **End date:** *15 April, 2025*			
Leader:	**Mustafa Berkan Yıkılmaz**	**Members involved:**	**Alper Göçmen, Mehmet Dedeler**
Objectives: ***Configure and maintain server infrastructure to ensure high availability, scalability, and security of the Article-Lens platform. This package focuses on the deployment and operational management of the server environment.***			
Tasks: ***Task 8.1 Server Configuration: Set up and configure server environments for development, testing, and production.*** ***Task 8.2 Performance Monitoring and Optimization: Implement tools and practices to monitor server performance and optimize resource usage.***			
Deliverables ***D8.1: Server setup documentation.*** ***D8.2: Performance reports and optimization guidelines.***			

WP 9: *Demo and Presentation*			
Start date: *11 December, 2024* **End date:** *15 May, 2025*			
Leader:	**Eray Yapağcı**	**Members involved:**	**All members**
Objectives: ***Prepare and present a comprehensive demonstration of the Article-Lens project to stakeholders, showcasing its capabilities and benefits.***			
Tasks: ***Task 9.1 Demo Preparation: Prepare all necessary materials and scripts for the demonstration.*** ***Task 9.2 Final Presentation: Execute a flawless presentation to stakeholders.***			
Deliverables ***D9.1: Demo materials.*** ***D9.2: Presentation feedback and adjustments report.*** ***D9.3: Analysis and Requirements report.***			

WP 10: ***Testing***			
Start date: ***01 February, 2024*** **End date:** ***15 May, 2025***			
Leader:	**Serra Çapraz**	**Members involved:**	**All members**
Objectives: ***Conduct thorough testing of all system components to ensure functionality, reliability, and user-friendliness under various scenarios.***			
Tasks: ***Task 10.1 System Testing: Perform comprehensive system testing, including unit, integration, and system tests.*** ***Task 10.2 User Acceptance Testing: Conduct user acceptance testing to ensure the system meets user expectations and requirements.*** ***Task 10.3 Domain User Feedbacks: Perform some tests with researchers and academicians to get domain-specific feedback to see possible errors and come up with proper updates.***			
Deliverables ***D10.1: Test plans and reports.*** ***D10.2: User acceptance test results.***			

WP 11: ***Beta Release & Project Launch***			
Start date: ***01 April, 2024*** **End date:** ***15 May, 2025***			
Leader:	**Mehmet Dedeler**	**Members involved:**	**All members**
Objectives: ***Launch a beta version of Article-Lens for selected users to gather early feedback and make necessary adjustments before the full public release.***			
Tasks: **Task 11.1 Beta Release Preparation: Prepare the system for beta release, ensuring all features are stable and documentation is complete.** **Task 11.2 Public Launch: Officially launch Article-Lens to the public, monitoring initial usage and feedback for rapid response to any issues.**			
Deliverables ***D11.1: Beta version of the platform.*** ***D11.2: Launch report and initial user feedback.***			

4.4 Ensuring Proper Teamwork

Effective teamwork is essential for the successful completion of the **Article-Lens** project. With multiple interconnected components and a diverse set of skills required, the team must adopt strategies to ensure collaboration, accountability, and seamless communication throughout the development lifecycle. The following measures will be implemented to promote proper teamwork:

Division of Responsibilities

- Tasks are divided into distinct work packages (e.g., frontend development, backend integration, testing) based on individual skills and expertise.
- Each team member is assigned clear responsibilities, ensuring ownership of specific tasks while promoting accountability.

Agile Development Approach

- The team follows an Agile methodology, using iterative sprints to deliver incremental improvements and regularly review progress [18].
- Weekly sprint planning sessions ensure tasks are appropriately prioritized, timelines are met, and potential blockers are addressed promptly.

Tools for Collaboration

- **YouTrack**: Used for task tracking and sprint management to keep the project organized and transparent [24].
- **GitHub**: Facilitates version control, enabling multiple team members to work on different parts of the codebase without conflicts [25].
- **Google Docs**: Maintains shared meeting notes and documentation for easy reference and collaboration [26].
- **WhatsApp**: Ensures asynchronous communication for quick updates and issue resolution [27].
- **Zoom**: Enables regular virtual meetings for progress reviews, brainstorming, and decision-making [28].

Regular Communication

- Weekly team meetings are scheduled to discuss progress, challenges, and next steps.
- Ad hoc meetings are held as needed to address critical issues or changes in project scope.

By fostering a collaborative environment and utilizing structured tools and processes, the team ensures efficient coordination, effective problem-solving, and the successful delivery of the **Article-Lens** project.

4.5 Ethics and Professional Responsibilities

The **Article-Lens** project strongly emphasizes ethical considerations and professional responsibilities to ensure the platform adheres to the highest standards of integrity, fairness, and accountability. These principles guide the development and deployment of the system, focusing on protecting user data, promoting fairness, and fostering trust among its users.

Data Privacy and Security

- **User Anonymity**: All user data, including search queries and preferences, will be anonymized before storage and analysis to protect individual privacy.
- **Encryption**: Sensitive information will be encrypted during transmission and at rest to safeguard against unauthorized access.
- **Compliance**: The platform will comply with global and regional data protection laws, including GDPR and KVKK, ensuring lawful data processing and storage [7][8].

Fairness and Transparency

- **Algorithmic Fairness**: The ranking algorithms will be designed to minimize biases, ensuring equitable representation of authors, institutions, and research topics.
- **Transparency**: Users will have access to information on how scores and rankings are calculated, fostering trust in the system.

Accountability in Development

- **Quality Assurance**: Rigorous testing protocols, including unit tests, integration tests, and user acceptance tests, will be implemented to ensure system reliability.
- **Professional Conduct**: Developers will adhere to established coding standards (e.g., PEP 8 for Python) and maintain comprehensive documentation for future maintainability.

Ethical Use of AI

- **Responsible AI Deployment**: The use of LLMs for summarization and ranking will be monitored to prevent misuse, such as generating misleading summaries or amplifying biases.
- **Transparency in AI Outputs**: Users will be informed that AI generates summaries and rankings and may require critical evaluation.

Commitment to User Empowerment

- **User Control**: The system will allow users to customize ranking criteria and summary formats, giving them autonomy over their research workflow.
- **Inclusive Design**: Accessibility features will be incorporated to support users with diverse needs, including those with disabilities.

By addressing these ethical considerations and professional responsibilities, **Article-Lens** aims to build a platform that meets functional and technical requirements and upholds the trust and confidence of its user base.

4.6 Planning for New Knowledge and Learning Strategies

The **Article-Lens** project requires the team to acquire new technical skills and domain knowledge to ensure the successful development and implementation of the platform. This section outlines the strategies for gaining new knowledge, addressing skill gaps, and ensuring continuous learning throughout the project lifecycle.

Identified Learning Areas

- **Large Language Models (LLMs)**: Understanding and fine-tuning LLMs for text summarization and ranking.
- **Django and REST API Development**: Building scalable and efficient backend services [19][29].
- **React for Frontend Development**: Designing responsive and interactive user interfaces [14].
- **PostgreSQL**: Setting up and optimizing a relational database for efficient data storage and retrieval [13].
- **Data Privacy Regulations**: Familiarize the team with GDPR and KVKK to ensure compliance [7][8].
- **User Experience Design**: Implementing accessibility and usability standards like WCAG 2.1 [5].

Strategies for Gaining Knowledge

- **Workshops and Online Tutorials**:
    - Participate in structured online courses on platforms like Coursera, Udemy, and YouTube for tools such as React, Django, and PostgreSQL [13][14][19][30][31][32].
    - Leverage tutorials and documentation provided by technology providers (e.g., OpenAI for LLMs, Django for backend development).
- **Hands-on Practice**:
    - Use test environments to implement and experiment with APIs, frontend components, and database schemas.
    - Collaborate on small sub-projects within the team to reinforce practical skills.
- **Peer Learning**:
    - Share expertise within the team by conducting knowledge-sharing sessions.
    - Assign learning mentors for team members with expertise in specific tools or technologies.
- **Consultations with Experts**:
    - Seek guidance from faculty advisors and industry mentors for challenges related to AI, data privacy, and software design.
    - Attend webinars or meetups on AI ethics and academic research tools.

Continuous Learning and Adaptation

- **Feedback Loops**:
    - Regularly incorporate feedback from beta testing and stakeholder reviews to identify areas for improvement.
    - Use this feedback to refine skills and address technical deficiencies.

- **Documentation and Knowledge Sharing**:
    - Maintain comprehensive internal documentation to support future learning and reduce onboarding time for new team members.
    - Archive lessons learned and key decisions for continuous improvement.

By adopting these strategies, the team ensures that it is well-equipped to handle the technical, ethical, and operational challenges of the **Article-Lens** project, fostering a culture of growth and innovation.

5. Glossary

API (Application Programming Interface): A set of rules and protocols that allow different software applications to communicate with each other [33]. This project uses APIs like arXiv and Google Scholar to retrieve academic data [1][4].

Django: A Python-based web framework used to build the backend of the **Article-Lens** platform, enabling the creation of RESTful APIs and database interactions [19].

GDPR (General Data Protection Regulation): A European Union regulation designed to enhance data protection and privacy for individuals. It ensures user data is handled securely and ethically [8].

KVKK (Kişisel Verilerin Korunması Kanunu):
Turkey's data protection law, similar to GDPR, focuses on the secure handling and processing of personal data [7].

LLM (Large Language Model): A type of artificial intelligence model trained on vast amounts of text data to perform language-related tasks such as summarization, translation, and ranking [34].

PostgreSQL: An open-source relational database management system that stores and queries user preferences, metadata, and other platform data [13].

React: A JavaScript library for building user interfaces utilized in the **Article-Lens** project for creating a responsive and interactive frontend [14].

RESTful API: An architectural style for designing networked applications, ensuring efficient and standardized communication between the frontend and backend components of the system [29].

UML (Unified Modeling Language): A standardized modeling language used to create diagrams, such as class diagrams and sequence diagrams, to visualize the system's architecture and behavior [35].

WCAG 2.1 (Web Content Accessibility Guidelines): A set of guidelines for making web content accessible to people with disabilities, ensuring inclusivity in the design of the **Article-Lens** platform [5].

6. References

[1] Google Scholar, “Google Scholar,” *Google.com*, 2024. https://scholar.google.com/

[2] Scholarcy, “Scholarcy”, *Scholarcy.com*. https://www.scholarcy.com/ (accessed Nov. 16, 2024).

[3] Research Rabbit, “ResearchRabbit,” *ResearchRabbit*, 2023. https://www.researchrabbit.ai/

[4] arXiv, "arXiv.org e-Print archive," 2024. [Online]. Available: https://arxiv.org/. [Accessed: Dec. 15, 2024].

[5] W3C, "Web Content Accessibility Guidelines (WCAG)," 2024. [Online]. Available: https://www.w3.org/WAI/standards-guidelines/wcag/. [Accessed: Dec. 15, 202

[6] ISO/IEC 27001: Information security, cybersecurity, and privacy protection — Information security management systems — Requirements," ISO, [Online]. Available: https://www.iso.org/standard/27001. [Accessed: Nov. 20, 2024].

[7] Personal Data Protection Authority," KVKK, [Online]. Available: https://www.kvkk.gov.tr/. [Accessed: Nov. 20, 2024].

[8] General Data Protection Regulation (GDPR) – Official Legal Text," GDPR Info, [Online]. Available: https://gdpr-info.eu/. [Accessed: Nov. 20, 2024].

[9] Google, "Google Chrome - Fast, secure, and free web browser," 2024. [Online]. Available: https://www.google.com/chrome/. [Accessed: Dec. 15, 2024].

[10] Mozilla, "Firefox Browser — Fast, private & free for Windows, Mac, Android, and iOS," 2024. [Online]. Available: https://www.mozilla.org/firefox/. [Accessed: Dec. 15, 2024].

[11] Apple, "Safari - Apple," 2024. [Online]. Available: https://www.apple.com/safari/. [Accessed: Dec. 15, 2024].

[12] Microsoft, "Microsoft Edge Browser," 2024. [Online]. Available: https://www.microsoft.com/edge. [Accessed: Dec. 15, 2024].

[13] PostgreSQL Global Development Group, "PostgreSQL: The world's most advanced open source database," 2024. [Online]. Available: https://www.postgresql.org/. [Accessed: Dec. 15, 2024].

[14] Meta, "React – A JavaScript library for building user interfaces," 2024. [Online]. Available: https://react.dev/. [Accessed: Dec. 15, 2024].

[15] Amazon Web Services, "Amazon Web Services (AWS) - Cloud computing services," 2024. [Online]. Available: https://aws.amazon.com/. [Accessed: Dec. 15, 2024].

[16] Google Cloud, "Google Cloud: Cloud computing services," 2024. [Online]. Available: https://cloud.google.com/. [Accessed: Dec. 15, 2024].

[17] GitLab, "GitLab CI/CD topics," 2024. [Online]. Available: https://about.gitlab.com/topics/ci-cd/. [Accessed: Dec. 15, 2024].

[18] K. Schwaber and J. Sutherland, "The Scrum Guide," 2024. [Online]. Available: https://scrumguides.org/. [Accessed: Dec. 15, 2024].

[19] Django Software Foundation, "The web framework for perfectionists with deadlines," 2024. [Online]. Available: https://www.djangoproject.com/. [Accessed: Dec. 15, 2024].

[20] Dublin Core Metadata Element Set, Version 1.1," Dublin Core Metadata Initiative, [Online]. Available: https://www.dublincore.org/specifications/dublin-core/dces/. [Accessed: Nov. 20, 2024].

[21] ISO 9241:Ergonomics of human-system interaction," ISO, [Online]. Available: https://www.iso.org/standard/77520.html. [Accessed: Nov. 20, 2024].

[22] OpenURL Standards," OCLC Research, [Online]. Available: https://www.oclc.org/research/activities/openurl.html. [Accessed: Nov. 20, 2024].

[23] Digital Object Identifier System," DOI, [Online]. Available: https://www.doi.org/. [Accessed: Nov. 20, 2024].

[24] JetBrains, "YouTrack: Project management for teams," 2024. [Online]. Available: https://www.jetbrains.com/youtrack/. [Accessed: Dec. 15, 2024].

[25] GitHub, "GitHub: Where the world builds software," 2024. [Online]. Available: https://github.com/. [Accessed: Dec. 15, 2024].

[26] Google, "Google Docs: Online document editor," 2024. [Online]. Available: https://www.google.com/docs/about/. [Accessed: Dec. 15, 2024].

[27] WhatsApp, "WhatsApp: Simple. Secure. Reliable messaging," 2024. [Online]. Available: https://www.whatsapp.com/. [Accessed: Dec. 15, 2024].

[28] Zoom, "Zoom Video Communications," 2024. [Online]. Available: https://zoom.us/. [Accessed: Dec. 15, 2024].

[29] Django Software Foundation, "Django REST framework: Web APIs for Django," 2024. [Online]. Available: https://www.django-rest-framework.org/. [Accessed: Dec. 15, 2024].

[30] Coursera, "Online Courses & Credentials from Top Educators," 2024. [Online]. Available: https://www.coursera.org/. [Accessed: Dec. 15, 2024].

[31] Udemy, "Online Courses - Learn Anything, On Your Schedule," 2024. [Online]. Available: https://www.udemy.com/. [Accessed: Dec. 15, 2024].

[32] YouTube, "YouTube," 2024. [Online]. Available: https://www.youtube.com/. [Accessed: Dec. 15, 2024].

[33] Amazon Web Services, "What is an API?" 2024. [Online]. Available: https://aws.amazon.com/tr/what-is/api/. [Accessed: Dec. 15, 2024].

[34] Cloudflare, "What is a large language model (LLM)?" 2024. [Online]. Available: https://www.cloudflare.com/learning/ai/what-is-large-language-model/. [Accessed: Dec. 15, 2024].

[35] Visual Paradigm, "What is UML?" 2024. [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/. [Accessed: Dec. 15, 2024].