

**National University of Sciences and Technology**  
**School of Electrical Engineering and Computer Science**



**Knowledge Verse**

**Final Year Project Report by**

Waleed Ahmed Shahid      283290

Hafiz Haseeb Ahmad Butt    313193

Sana Ullah                  294942

**Supervisor:**

Dr. Shah Khalid

**Co-Supervisor:**

Dr. Hasan Ali Khattak

**Bachelor of Science in Computer Science (2019-2023)**

Department of Computing

School of Electrical Engineering and Computer Science

National University of Sciences & Technology

## **DECLARATION**

We hereby declare that this project report entitled “KnowledgeVerse” submitted to the School of Electrical Engineering and Computer Science, is a record of an original work done by us under the guidance of Supervisor Dr. Hasan Ali Khattak and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Computer Science.

### **Team Members**

### **Signature**

Waleed Ahmed Shahid

Hafiz Haseeb Ahmad Butt

Sana Ullah

### **Advisor**

### **Signature**

Dr. Shah Khalid

### **Co-Advisor**

### **Signature**

Dr. Hassan Ali Khattak

## **DEDICATION**

We dedicate our dissertation work to Allah Almighty, To our family, friends, and advisors. We will always appreciate all that they have done in supporting and guiding us to achieve the milestone of completing our work.

## **ACKNOWLEDGEMENTS**

All glory to Allah Almighty who gave us the strength and determination to work on the project. Special thanks to Dr. Shah Khalid for being patient and constant guidance throughout our work. Without their help and resources this would not have been possible and for that we are forever grateful.

## Table of Contents

List of Figures and Diagrams: .....	8
List of Class Diagram .....	8
List of Activity Diagrams.....	8
List of Sequence Diagrams .....	8
List of User Interfaces .....	8
List of Tables.....	9
List of Schema Tables .....	9
List of Test Cases .....	10
<b>1. Background:</b> .....	11
<b>1.1. Problem Definition:</b> .....	11
1.1.1. Problem Statement: .....	12
<b>1.2. Literature Review:</b> .....	12
1.2.1. Paper-1 .....	12
1.2.2. Paper-2 .....	13
1.2.3. Paper-3 .....	14
<b>1.3. Business Opportunity:</b> .....	15
<b>1.4. Proposed Solution:</b> .....	15
<b>1.5. Project Scope:</b> .....	16
<b>1.6. Stakeholders:</b> .....	16
<b>1.7. Tech Stack Used:</b> .....	17
<b>2. Software Requirements Specification:</b> .....	18
<b>2.1. Requirement Elicitation Process:</b> .....	18
2.1.1. Users Requirement:.....	18
2.1.2. Functional Requirements:.....	18
2.1.3. Non-functional Requirements:.....	18
<b>2.2. Use Case Diagrams:</b> .....	19
<b>2.3. Functional Requirements:</b> .....	20
2.3.1. Contextual Analysis .....	20
2.3.2. Knowledge Graph & Citation Network.....	21
2.3.3. Freshness of Documents .....	22

2.3.4. Filters.....	22
2.3.5. 3D Graph Display.....	23
2.3.6. Recommendation Engine .....	23
2.3.7. Sign Up .....	24
2.3.8. Log-In & Log-Out .....	25
2.3.9 Add & Delete Papers .....	25
2.3.10. View Stats.....	26
2.3.11 Manage Authors.....	26
<b>2.4. Non-functional Requirements .....</b>	<b>27</b>
2.4.1. Performance Requirements .....	27
2.4.2. Security Requirements .....	27
2.4.3. Software Quality Attributes.....	27
<b>3. Software Design Specification.....</b>	<b>29</b>
<b>3.1. Architectural Design .....</b>	<b>29</b>
3.1.1. User Interface Component (View):.....	29
3.1.2. Business Logic Component (Controller): .....	29
3.1.3. Data Logic Component (Model): .....	30
<b>3.2. Design Methodology.....</b>	<b>31</b>
3.2.1. Encapsulation.....	31
3.2.2. Inheritance.....	31
3.2.3. Modularity.....	31
<b>3.2.4. Software Process Model .....</b>	<b>32</b>
3.2.5. Flexibility .....	32
3.2.6. Faster time-to-market.....	32
3.2.7. Improved collaboration .....	32
3.2.8. Increased transparency .....	32
3.2.9. Better risk management .....	32
<b>3.3. Data Representation Diagram:.....</b>	<b>33</b>
3.3.1. Entity Relationship Diagram:.....	33
3.3.2. Class Diagram:.....	34
<b>3.4. Process Flow: .....</b>	<b>35</b>
3.4.1. Activity Diagrams .....	35

<b>3.5. Design Models .....</b>	<b>39</b>
3.5.1. Sequence Diagrams.....	39
3.5.2. Data Dictionary.....	43
<b>4. System Testing .....</b>	<b>50</b>
<b>4.1. Functional testing:.....</b>	<b>50</b>
4.1.1. Unit testing:.....	50
4.1.2. Integration Testing:.....	64
<b>4.2. Non-functional Testing: .....</b>	<b>66</b>
4.2.1. Performance Testing: .....	66
4.2.2. Security Testing .....	67
<b>5. System Integration and Deployment:.....</b>	<b>68</b>
<b>5.1. System Integration .....</b>	<b>68</b>
5.1.1. Email Integration:.....	69
<b>5.2. System Deployment: .....</b>	<b>69</b>
<b>6. Key User Interfaces.....</b>	<b>70</b>
Home Page:.....	70
Search Page:.....	72
Single Paper Page:.....	73
Contact Page: .....	74
Author Profile Page: .....	75
Sign Up: .....	76
Sign In: .....	77
Forgot Password:.....	78
Change Name:.....	78
Publisher account Settings:.....	79
Admin Interfaces:.....	84
<b>7. Conclusion and Future Scope .....</b>	<b>88</b>

## List of Figures and Diagrams:

Figure 1: Tech Stack .....	17
Figure 2: Use case diagram: .....	20
Figure 3: System Architecture .....	30

## List of Class Diagram

Figure 4: Class Diagram .....	35
-------------------------------	----

## List of Activity Diagrams

Figure 5: Activity Diagram - User Sign Up .....	36
Figure 6: Activity Diagram -Log-In Activity Diagram .....	36
Figure 7: Activity Diagram -Publisher/Author Activities .....	37
Figure 8: Activity Diagram - User Search.....	37
Figure 9: Activity Diagram -Admin .....	38

## List of Sequence Diagrams

Figure 10: Profile Authentication Sequence Diagram.....	39
Figure 11: Sequence Diagram for User Search .....	40
Figure 12: Sequence Diagram Functionalities Provided to Researcher for updating its own profile.....	41
Figure 13: Figure-4 Carried .....	42
Figure 14: Sequence Diagram for Admin Functionalities.....	42
Figure 15: Figure-6 Carried .....	43

## List of User Interfaces

Figure 16: User Interface: Home Page .....	71
Figure 17: User Interface - Search Page .....	73
Figure 18: User Interface - Single Paper.....	74

Figure 19: User Interface - Contact Page .....	75
Figure 20: User Interface - Author Profile.....	76
Figure 21: User Interface- Sign Up page .....	77
Figure 22: User Interface - Sign In.....	77
Figure 23: User Interface - Change Password .....	78
Figure 24: User Interface - User account settings.....	79
Figure 25: User Interface - Researcher Information .....	80
Figure 26: User Interface - Publisher Articles .....	81
Figure 27: User Interface - Published Articles Details.....	82
Figure 28: User Interface - Publisher Profile .....	83
Figure 29: User Interface - Publisher Requests.....	84
Figure 30: User Interface - Articles Request .....	85
Figure 31: User Interface - Rejected Articles .....	86
Figure 32: User Interface- Admin Statistics.....	87

## List of Tables

Table 1: Stakeholder's Table.....	16
-----------------------------------	----

## List of Schema Tables

Table 2: Schema Table -User Table.....	43
Table 3: Schema Table -Publisher Table .....	44
Table 4: Schema Table -Admin Table.....	44
Table 5: Schema Table - Affiliations Table .....	45
Table 6: Schema Table -Areas of Interest Table.....	45
Table 7: Schema Table -Author Names Table.....	45
Table 8: Schema Table - Blocked Authors Table .....	46
Table 9: Schema Table - Articles Table .....	46
Table 10: Schema Table -References Table.....	47
Table 11: Schema Table - CoAuthors Table .....	47
Table 12: Schema Table -Saved Articles Table .....	47
Table 13: Schema Table - Article Topics Table .....	48
Table 14: Schema Table -Research Topics Table .....	48
Table 15: Schema Table - Added Articles Table.....	48
Table 16: Schema Table - Deleted Articles Table.....	49

Table 17: Schema Table - Journal Table.....	49
---	----

## List of Test Cases

Table 18: Test Case for User/Publisher Sign Up .....	51
Table 19: Test case for user/publisher login.....	52
Table 20: Test case for Log out .....	54
Table 21: Test case for Publisher Options .....	54
Table 22: Test cases for Articles Uploading .....	55
Table 23: Test case for Citation Graph Display .....	57
Table 24: Test case for save articles .....	57
Table 27: Test case for Query search.....	58
Table 28: Test case for filtered search.....	59
Table 31: Test case for Related Articles .....	60
Table 32: Test case for cited articles.....	61
Table 29: Test case for publisher profile.....	61
Table 30: Test case for Admin stats .....	62
Table 24: Test case for Managing Authors .....	62
Table 26: Test case for managing articles.....	63

# **Chapter 1**

## **1. Background:**

Pakistan has traditionally lacked a comprehensive scholarly search engine, which has led to difficulties for researchers and academics in accessing and finding relevant academic literature. While there are some existing search engines, many lack accurate ranking features and effective visualization tools for search results. As a result, there is a need for a more efficient and user-friendly platform for finding and accessing academic literature in Pakistan. In order to address this gap, our team has developed KnowledgeVerse, a sophisticated search platform that offers advanced filtering options, relevance ranking, and powerful visualization tools to help scholars find the most relevant and up-to-date research literature. We have tried to rank the papers by some new feature, i.e., freshness, which wasn't being used commonly in the ranking of scholar papers. We have added filtered search in our search engine which will enable users to give more priority to some particular feature which wasn't available before and no control was available to user.

### **1.1. Problem Definition:**

The academic search platforms available in the market are not equipped with sufficient features and ranking models to deliver the most relevant content that users are looking for. This has become a major issue as the number of scholarly publications is rapidly increasing, and researchers are struggling to find the papers that are most relevant to their needs. The current ranking models are not effectively addressing this issue, and there is a need for a more robust and efficient approach to scholarly search.

Additionally, the lack of a dedicated academic search engine owned by Pakistan is another major issue that Pakistani scholars are facing. As a result, they have to rely on foreign academic search platforms, which often require payment for access to relevant research papers. This puts a financial burden on Pakistani scholars and creates a barrier to accessing the information they need for their research.

Therefore, there is a pressing need for an academic search platform that is not only equipped with advanced ranking models to deliver the most relevant content but is also owned and operated in Pakistan to provide cost-effective access to scholarly publications. This will not only benefit Pakistani scholars but will also contribute to the growth and development of the academic community in the country.

#### 1.1.1. Problem Statement:

“Researchers use scholarly search engines that lack enough features to rank papers effectively.”

### **1.2. Literature Review:**

For this project, a comprehensive literature review was conducted to gain a deeper understanding of the current state-of-the-art in scholarly search engines. This review was essential in identifying the key challenges and opportunities in developing a new search platform for academic literature in Pakistan.

Numerous research papers were analyzed to understand the latest techniques and algorithms being used in scholarly search engines, with a particular focus on relevance calculation, freshness, citation score, and recommendation engines. This provided valuable insights into the strengths and weaknesses of existing platforms and helped guide the development of our search engine.

In addition to academic literature, we also consulted with experts in the field and studied the user feedback on existing search engines to better understand the user requirements and pain points. This helped in developing an effective filtering system to narrow down search results and a user-friendly interface to enhance the user experience.

Overall, the literature review proved to be a crucial component of the project, providing valuable insights and direction for the development of the Scholarly Search Engine. Some of the research papers and their descriptions are given below:

#### 1.2.1. Paper-1

“BERT-Embedding and Citation Network Analysis based Query Expansion Technique for Scholarly Search”

**Description:**

The proliferation of research publications has posed a challenge to academic search engines to provide the most relevant papers to users. Researchers have proposed various solutions over the years, such as query expansion and citation analysis, to improve the effectiveness of academic search engines. However, query expansion techniques may introduce non-relevant information while expanding the original query. Recently, the BERT model has shown success in contextualized document retrieval.

In response to the challenges faced by academic search engines, this paper proposes a novel approach called QeBERT. The approach combines BERT-based embedding and Citation Network Analysis (CNA) in query expansion to improve scholarly search. Specifically, QeBERT uses the context-aware BERT-embedding and CNA for query expansion in Pseudo-Relevance Feedback (PRF) fashion.

Initial experimental results on the ACL dataset show that BERT-embedding can provide a valuable augmentation to query expansion and improve search relevance when combined with CNA. This approach has the potential to improve the effectiveness of academic search engines and reduce the time and effort required by researchers to find relevant papers.

### 1.2.2. Paper-2

“A multi-objective approach to determining the usefulness of papers in academic search.”

**Description:**

The effectiveness of academic search engines has been a key concern in the research community. However, previous investigations have mostly focused on the relevancy of papers to queries, without taking into consideration other aspects that may affect their usefulness, such as their publication age and impact in the research community. To address this issue, this paper proposes a framework that considers relevance, publication age, and impact scores to rank academic papers. A set of metrics is introduced to measure the usefulness of a ranked list of papers, and the framework is evaluated using the ACL dataset. The results show that the proposed ranking algorithm is effective in improving usefulness when two or three aspects of

academic papers are considered simultaneously. The proposed multi-objective academic search framework is the first of its kind and is evaluated with a set of new evaluation metrics, making it a significant contribution to the field.

### 1.2.3. Paper-3

“An Effective Scholarly Search by Combining Inverted Indices and Structured Search With Citation Networks Analysis”

#### **Description:**

The exponential growth of scholarly publications in various digital platforms has made it difficult for researchers to find relevant research papers. While many retrieval systems, such as Google Scholar, Semantic Scholar, PubMed, and CiteSeerX, have attempted to mitigate this problem, there is still a lot of room for improvement in meeting the scholarly needs of users. Previous studies have employed different techniques to address this issue, including classical Information Retrieval (IR) and semantics-aware methods like Knowledge Graph (KG). However, it is hypothesized that a combination of both approaches can further improve search relevance.

In this regard, this paper proposes a solution that combines inverted index from classical IR with BM25 as the weighting scheme, along with Citation Networks Analysis (CNA), to generate initial search results. The top-k entities from the initial search results are then used as a search query to the KG to re-rank the results. This approach allows for the exploitation of both textual content and structural semantics of research publications in the retrieval process. The aim is to gain insights into the behavior of both textual and structured information in the strategic ranking of scholarly articles.

The proposed technique has been evaluated using the ACL Anthology Network (AAN) dataset. The results demonstrate that the approach can improve retrieval performance in terms of Normalized Discounted Cumulative Gain (nDCG) and precision rates. By combining classical IR and KG-based retrieval techniques, the proposed solution aims to provide a more effective and efficient scholarly search engine for researchers.

### **1.3. Business Opportunity:**

Our project, KnowledgeVerse, presents a unique business opportunity due to its potential to address the unmet needs of scholars and researchers worldwide. By providing a user-friendly platform that utilizes advanced machine learning algorithms and data visualization tools, we can revolutionize the way research is conducted.

With the growing demand for access to scholarly information, KnowledgeVerse can provide a cost-effective solution that offers relevant and up-to-date information to users worldwide. By targeting the Pakistani market, where there is a lack of locally owned scholarly search engines, we can create a strong customer base while also expanding globally.

The revenue model of KnowledgeVerse can include subscription-based access to premium content, targeted advertising, and partnerships with academic institutions and publishers. Additionally, the platform can offer value-added services such as personalized recommendations, collaboration tools, and citation analysis, further increasing its potential for revenue generation.

Overall, KnowledgeVerse has the potential to become a leading platform for scholarly research and offer significant business opportunities in the rapidly growing digital marketplace.

### **1.4. Proposed Solution:**

KnowledgeVerse is an innovative platform designed to revolutionize the way we search for scientific literature. As researchers, academics, and students, we all face the challenge of finding the most relevant and up-to-date papers for our work. While there are many search engines available, the process of locating a relevant paper can still be time-consuming and frustrating. Our solution to this challenge is KnowledgeVerse, a search platform built on advanced algorithms that can find and retrieve the most relevant scientific literature quickly and efficiently. The platform offers a range of filtering options such as author and freshness ranking to help users find the most suitable articles.

KnowledgeVerse indexes academic literature from a variety of sources, including academic journals, institutional repositories, and academic databases. The platform is designed to be scalable, allowing for the addition of new features and functionality in the future. The platform also allows scholars to submit their articles for indexing and ranking. This feature ensures that the most recent and cutting-edge research is available to users, thereby promoting the growth and advancement of knowledge in various fields.

## **1.5. Project Scope:**

The project scope for our project, KnowledgeVerse, is to develop a web application that utilizes machine learning algorithms and natural language processing techniques to provide users with relevant and personalized scholarly search results. The web application will be accessible to users worldwide and will be able to retrieve information from various scholarly sources. The focus will initially be on the Pakistani audience, but the project has the potential to expand globally in the future.

The scope of the project includes the development of a user-friendly interface, the implementation of machine learning algorithms for personalized search results, and the integration of natural language processing techniques for improved search queries. Additionally, the project will involve the construction of a large-scale dataset and the use of citation networks analysis to support scholarly search.

## **1.6. Stakeholders:**

Our project can be used in multiple ways, first of all it can be delivered totally as a project, and it can also be used to make small modules for different search features in different projects. The main stakeholders of the project are as follows.

*Table 1: Stakeholder's Table*

<b>Stakeholders</b>	<b>Description</b>
Publishers	Authors who want to publish their data

Scholars	Users who want to read data
Search	People who want to buy our customized algorithm for context relevant search

## 1.7. Tech Stack Used:

The technologies which we are using in this project are as follows:

- React Js (Front-End)
- Spring Boot(Back-End)
- Databases( Sql, Solr)
- Graph database(Neo4j)
- Python (NLP)

Figure 1: Tech Stack



# **Chapter 2**

## **2. Software Requirements Specification:**

### **2.1. Requirement Elicitation Process:**

Requirement elicitation is a crucial process in software engineering that involves collecting and defining the functional and non-functional requirements of a project. It is the process of identifying and documenting the needs of the stakeholders and the end-users for the software system.

#### **2.1.1. Users Requirement:**

The user requirements for KnowledgeVerse include a simple and intuitive user interface that allows users to easily search for scholarly content. Users should be able to search for papers by keyword, author and by different features. Additionally, the system should be accessible to users from a variety of devices, including desktop computers, laptops, tablets, and smartphones. Finally, the system should provide a secure and reliable platform for users to access and download scholarly material, with appropriate measures in place to protect user data and ensure the integrity of the content.

#### **2.1.2. Functional Requirements:**

Functional requirements for KnowledgeVerse include the ability to search and browse scholarly articles, papers, and other research material based on various filters such as keywords, authors, publication dates, and journals. Additionally, it should have the functionality to suggest related articles and recommend articles based on the user's selected paper and reading behavior. Another important functionality is that users can view citation network in graph form.

#### **2.1.3. Non-functional Requirements:**

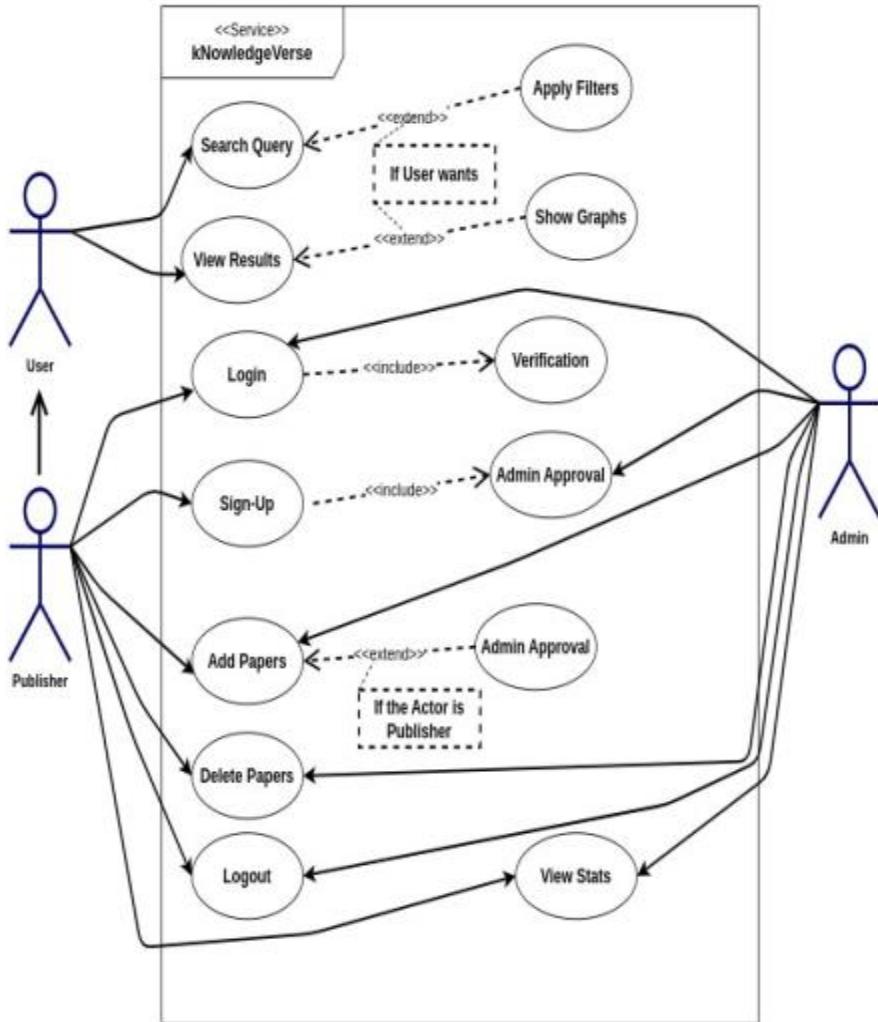
Non-functional requirements are the platform's technical specification such as performance, reliability etc. Our project KnowledgeVerse has certain performance requirements that we need to meet to compete with other search platforms. The results should be calculated within a

maximum of one second, and the UI should be easy to load and work efficiently on all devices regardless of the platform. The system should always be available, and in case of hardware failure or database corruption, a replacement page will be shown, and backups of the database should be retrieved. It is essential that the system can handle a substantial number of users at the same time, particularly during peak hours. In terms of security requirements, user profiles and other data available to us should be protected and secured, and authentication should be required for each user.

## **2.2. Use Case Diagrams:**

The use case diagram for KnowledgeVerse would include three actors: User, Publisher, and Admin. The User can search for a query and view different papers that match the search criteria. The User can then select the best paper based on the provided information. The Publisher inherits the User's role and can also upload papers to the platform. The Admin has complete control over the platform and can manage the publishers' status, update papers, and view statistics related to the platform's performance. The use case diagram would also include various use cases such as searching for papers, uploading papers, updating publisher status, and viewing statistics, which would be associated with their respective actors. Overall, the use case diagram for KnowledgeVerse would provide a clear overview of the different actors, their roles, and the interactions they have with the system.

Figure 2: Use case diagram:



## 2.3. Functional Requirements:

The functional requirements of our project are mentioned below:

### 2.3.1. Contextual Analysis

#### Description and Priority

Considering that most search engines do not utilize contextual analysis of scientific literature to determine the relevance of an item to a query. This feature is being used to determine how

contextually relevant the papers are. The primary characteristic of our platform that sets it apart from other similar platforms is this.

### **Stimulus/Response Sequences**

When a user enters a query, our back end processes it to locate documents that are contextually relevant using the Solr Engine, which uses a variety of criteria to help the ML-model for calculating the contextual relevance between the user query and content.

### **Functional Requirements**

REQ-1: Algorithm should calculate contextual relevance between query and papers.

REQ-2: Should interpret query with respect to its context.

REQ-3: Find data based on contextual analysis not word-based analysis.

### **2.3.2. Knowledge Graph & Citation Network**

#### **Description and Priority**

This is the second element that distinguishes us from other platforms of a similar nature. To determine the appropriate documents, we are using the graph information of our material. The right ranking will be determined by the ML-Model using the score derived from these graphs.

### **Stimulus/Response Sequences**

When a user enters a query, our back end processes it to locate documents that are contextually relevant using the Solr Engine, which uses Graph Knowledge to help ML-model for calculating the contextual relevance between the user query and content.

### **Functional Requirements**

REQ-1: Calculate graph-based scores of each paper returned from contextual relevant model.

REQ-2: These scores will be calculated using, the standard formulas already provided in Neo4J database.

REQ-3: Should calculate graph-based scores correctly and efficiently.

### 2.3.3. Freshness of Documents

#### **Description and Priority**

We are also giving importance to recent documents rather than just calculating the relevance, we are calculating the how recent is the document and giving it high importance. Recent documents will be of more importance than old documents.

#### **Stimulus/Response Sequences**

When a user enters a query, our back end processes it to locate documents that are contextually relevant using the Solr Engine, which uses Graph Knowledge along with freshness of paper to help ML-model for calculating the contextual relevance between the user query and content.

### **Functional Requirements**

REQ-1: Freshness of documents should be calculated using reciprocal of the difference between published years.

REQ-2: Giving good scores to fresh papers in ranking.

### 2.3.4. Filters

#### **Description and Priority**

Other platforms do not offer many filters for filtering the pertinent papers. Thus, we are offering a list of filters to assist users in finding the necessary documents. Finding papers authored by any group of writers is one of these criteria. Sorting by date or significance. Whether or not to incorporate the citations and other graph data.

#### **Stimulus/Response Sequences**

When a user enters the query, it can apply these filters to narrow down the results, which it wants to see.

### **Functional Requirements**

REQ-1: Whenever applied, the results should be modified accordingly.

REQ-2: There should be a default form of filter.

#### 2.3.5. 3D Graph Display

##### **Description and Priority**

No other platform provides the visual representation of different networks of documents on their website. We are providing the document network representation live on our website and it makes us different from other Software Requirements Specification for KnowledgeVerse Page 17 platforms. This visual representation will help the user to see and find the other documents and their importance on graph.

##### **Stimulus/Response Sequences**

When data is responded to front-end after calculating the relevant data, their graph is also provided to the front-end. We can build a live 3D graph using that information, on our website. Users just must select the option to see the results in graphical interface.

### **Functional Requirements**

REQ-1: 3D graph will consist of connected entities returned by the server.

REQ-2: User should be able to explore further data through 3D graph.

#### 2.3.6. Recommendation Engine

##### **Description and Priority**

We are also providing recommendations or related paper features on our website. User will be given related papers based on the paper he/she has selected from the retrieved papers.

### **Stimulus/Response Sequences**

For initial recommendations, users which have not selected any other paper can also get see some general recommendation papers based on searching a paper's title . For other users which have authored different publications, will receive recommendations based upon their topics of research.

### **Functional Requirements**

REQ-1: Collect user's interested topics, recent searches, and area of fields to determine user behavior.

REQ-2: Provide recommendations based on data collected from each user based on its search behavior and provide general recommendation based on current trends in searching behavior.  
of users.

REQ-3: Should work in parallel with our relevance model.

#### **2.3.7. Sign Up**

##### **Description and Priority**

Only admin can create new account of for publisher to see its information. These sign-up credentials will be shared with publisher then the publisher can access all the resources information it wants.

### **Stimulus/Response Sequences**

Since over data is gathered either through data exploration or by providing the subscription to any publisher. So, first data is stored in database and all its stats and other information is calculated. To give access to information we produced using data, publisher's account will be created and provided to the publisher.

### **Functional Requirements**

REQ-1: Admin should be able to create account of publisher.

REQ-2: Account will be created based on request from publisher.

### 2.3.8. Log-In & Log-Out

#### **Description and Priority**

Both the publisher and Admin should be able to log-in and log-out from their accounts.

#### **Stimulus/Response Sequences**

For Log-In purposes, authentication is required on both sides. When authentication is completed, both the publisher and administrator should be able to access their accounts.

#### **Functional Requirements**

REQ-1: Administrator & Publisher should be authenticated before access to their accounts.

### 2.3.9 Add & Delete Papers

#### **Description and Priority**

Administrator should be able to add and delete papers as per request from Publisher. Data can also be included through crawler automatically and will be deleted as per request from the Publisher or directly by admin too if there is an issue in the paper.

#### **Stimulus/Response Sequences**

Data will be added either automatically through crawler or on request from Publisher. Data will be deleted as per request from Publisher. As per the data is updated, the stats are also updated automatically.

#### **Functional Requirements**

REQ-1: Administrator should be able to add and delete the data, as per-request.

REQ-2: As data gets updated, the stats should also be updated regarding the data.

### 2.3.10. View Stats

#### **Description and Priority**

Both the Administrator and Publisher should be able to view stats. Administrator should be able to view stats of any data, they want. Publisher will only be able to see stats of its own account.

#### **Stimulus/Response Sequences**

Both Administrator and Publisher will log-in to their accounts and will be able to see stats as per the privilege.

#### **Functional Requirements**

REQ-1: Administrator should be able to see any stats, as they want.

REQ-2: Publisher should be able to see its stats after logging-in.

### 2.3.11 Manage Authors

#### **Description and Priority**

Administrator should be able to manage all users and Publisher. Admin can give privilege to any other publisher and can change the data, if any monopoly is monitored.

#### **Stimulus/Response Sequences**

Admin will be able to manage Publishers and other users. Admin will manage the data for each publisher and results will be displayed as per the management.

#### **Functional Requirements**

REQ-1: Administrator should be able to manage data of each Publisher.

REQ-2: Admin should also provide customer support help.

## 2.4. Non-functional Requirements

### 2.4.1. Performance Requirements

Our results should be calculated in a maximum of one second if we want to compete with other available search platforms. Other platforms use distributed computing and CDNs to perform efficiently and faster. Our UI should be easy and efficient enough to load faster to handle and work efficiently on all devices irrespective of the platform on which it is running. The system should always be available, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved.

This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time. The times when the system will be under the most stress is likely during peak hours.

### 2.4.2. Security Requirements

User profiles along with other data available to us should be protected and secured, make sure that our data is only available through our defined way, no other person/application should be able to access it without our permission. There should be authentication for each user trying to create an account on our platform. Confidentiality of our data should be assured.

- Passwords will be saved encrypted in the database to ensure the user's privacy.
- The user's IP will be logged.
- The system will be protected against vulnerabilities such as SQL injection attacks.

### 2.4.3. Software Quality Attributes

For Users, attributes of our product are:

- **Usability**

- o The System interface shall be easy to use and shall not require any user manuals.
- o The application shall display different messages to guide users.

- o The system shall be able to handle 10000 recognition requests per minute.

- **Correctness**

- o The system shall give accurate results based on the context of the query.

- o The system shall show results as requested by the user more than 90% of the time.

- o The “suggest me a paper” feature should display results if the relevancy of selected keywords is greater than 90%.

- **Availability**

- o System shall be up 24/7 and only have downtime of 30 minutes every month for maintenance.

- **Robustness**

- o A message shall be displayed to the user if any information is missing during searching.

For Developers, attributes of our product are:

- **Scalable Design & Architecture**

- o Developers can extend the system servers if the number of users on the software exceeds the normal limit on a daily basis.

- **Flexibility of Design**

- o This design is flexible, iterative, and emergent, therefore it can easily be changed and adapted to meet future needs by any future developer.

- **Code reusability**

- o We are developing different algorithms for ranking scholarly documents, so our algorithms or code may be useful for other purposes as well.

# **Chapter 3**

## **3. Software Design Specification**

### **3.1. Architectural Design**

We have followed MVC architecture in our project. So, there are three main components of our application, which are:

#### **3.1.1. User Interface Component (View):**

It is responsible for presenting the data to the user and capturing user input. Its main purpose is to provide an interface for the user to interact with the application and to display the information in a way that is easily understandable and usable.

It is implemented as graphical elements such as buttons, text fields, and drop-down menus, which allow the user to interact with the application. The design of the UI Layer is user-friendly and intuitive, allowing the user to easily navigate the application and complete tasks.

At this point we have provided further filters which are not present in other search platforms, allowing the user to minimize the results and easily find what they want.

#### **3.1.2. Business Logic Component (Controller):**

In Model-View-Controller (MVC) architecture, the Business Logic Layer (Controller) is responsible for coordinating the interactions between the Model and the View. Its main purpose is to receive user input from the View, update the Model with the input, and update the View with changes to the data.

The Controller is responsible for implementing the application's behavior in response to user actions. For example, it may initiate a request to retrieve data from the Model, or it may initiate a process to update the data in the Model.

In our application it receives requests from users and provide response to it, by calculating the results based on data available in our databases.

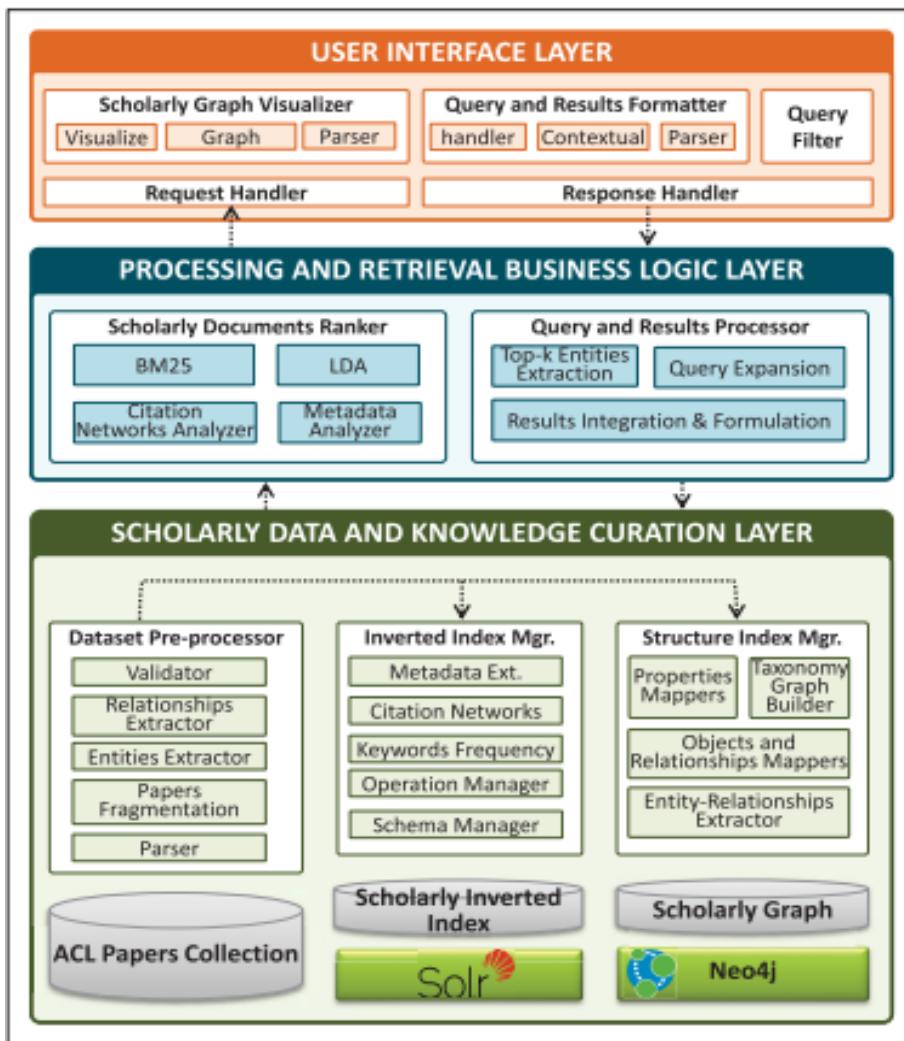
It should be designed in a way that separates the application's behavior from the data and the user interface, making it easy to maintain and modify the application.

### 3.1.3. Data Logic Component (Model):

The Model communicates with the Controller to update or retrieve data and to provide information about the state of the application. The Controller, in turn, communicates with the View to update the data being displayed and to provide user input.

The Model contains the data that the application uses and manipulates. In our project the model is implemented using Solr Engine (for textual relevance) & Neo4J (for graphical Storage) of data.

*Figure 3: System Architecture*



## **3.2. Design Methodology**

A scientific literature search engine is a complex system that requires the efficient management of a large amount of data and the ability to handle a wide range of user requests. It is therefore important to choose a design methodology that can effectively support the development of such a system.

Object-Oriented Programming (OOP) is a programming paradigm that emphasizes the concept of objects and classes and provides a way to encapsulate data and behavior into modular units. OOP provides several benefits for the development of complex systems like a scientific literature search engine, including:

### **3.2.1. Encapsulation**

OOP allows the encapsulation of data and behavior into objects, providing a clear separation between the data and the operations that can be performed on it. This makes it easier to maintain and test the code, as well as to extend or modify the functionality of the system.

### **3.2.2. Inheritance**

OOP provides inheritance, which allows objects to inherit characteristics from a parent class. This can be used to create a hierarchy of classes, with more specialized classes inheriting from more general ones. This can simplify the code and reduce redundancy.

### **3.2.3. Modularity**

OOP emphasizes the creation of modular units, making it easier to break down a complex system into smaller, more manageable parts. This can help improve the maintainability and scalability of the system.

This design would allow for the modularization of the code and easy extension or modification of the search engine's functionality in the future. OOP also provides a clear separation between the data and behavior of the system, making it easier to maintain and test the code.

In summary, the choice of OOP for a scientific literature search engine is driven by the need for a modular, flexible, and maintainable design that can handle the complex requirements of the system. OOP provides a clear and effective way to encapsulate data and behavior, inherit

characteristics from parent classes, and break down the system into modular units, making it an ideal choice for the development of a scientific literature search engine.

#### 3.2.4. Software Process Model

Software Process model refers to the software development life cycle (SDLC) model that the team follows. The most common process models are Waterfall, Agile, and DevOps.

The choice of process model depends on various factors, such as the size and complexity of the project, the level of stakeholder involvement, and the desired level of flexibility and adaptability.

Here are some reasons for choosing the Agile process model for our project:

#### 3.2.5. Flexibility

Agile methodology allows for frequent adaptation and modification of the project based on user feedback and changing requirements. This can be particularly useful in the development of an academic search engine where user requirements may evolve over time.

#### 3.2.6. Faster time-to-market

Agile emphasizes delivering a minimum viable product quickly, which can be beneficial for an academic search engine project where users may be eager for a functional solution.

#### 3.2.7. Improved collaboration

Agile promotes close collaboration between developers, stakeholders, and users, which can help ensure that the final product meets the needs of all parties involved.

#### 3.2.8. Increased transparency

Agile provides regular updates and demonstrations of working software, which can increase transparency and build trust with stakeholders.

#### 3.2.9. Better risk management

The iterative nature of Agile enables the project team to identify and mitigate potential risks early on, helping to ensure the success of the academic search engine project.

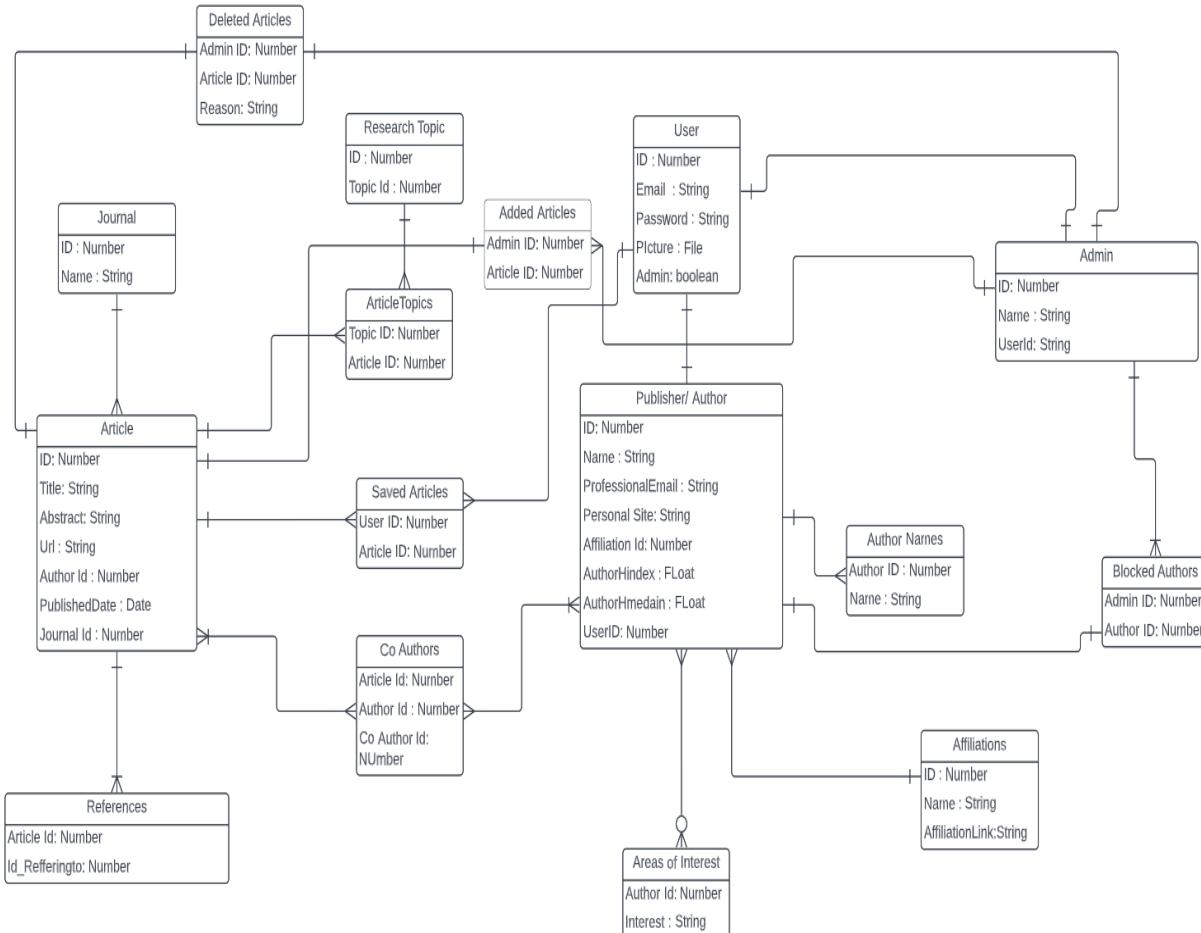
### **3.3. Data Representation Diagram:**

A data representation diagram is a graphical representation that shows the structure and relationships between different data entities in a system. It is used to represent the data schema or data model of a system, including the tables, columns, and relationships between them. The diagram typically shows the data entities as boxes or rectangles, with the attributes or columns of the entities listed inside them. The relationships between the entities are shown as lines or arrows connecting the boxes. Data representation diagrams are useful in database design and analysis.

#### **3.3.1. Entity Relationship Diagram:**

The Entity Relationship Diagram (ERD) for KnowledgeVerse provides a visual representation of the database schema and the relationships between its entities. The ERD is an essential tool for database design, as it helps to ensure data integrity and consistency. It shows the various entities in the database, their attributes, and how they are connected to each other. The ERD helps in understanding the relationships between the entities and how data flows through the system. It also makes it easy to identify potential problems or inconsistencies in the database design. The ERD for KnowledgeVerse is shown below.

Figure 2: Entity Relationship Diagram

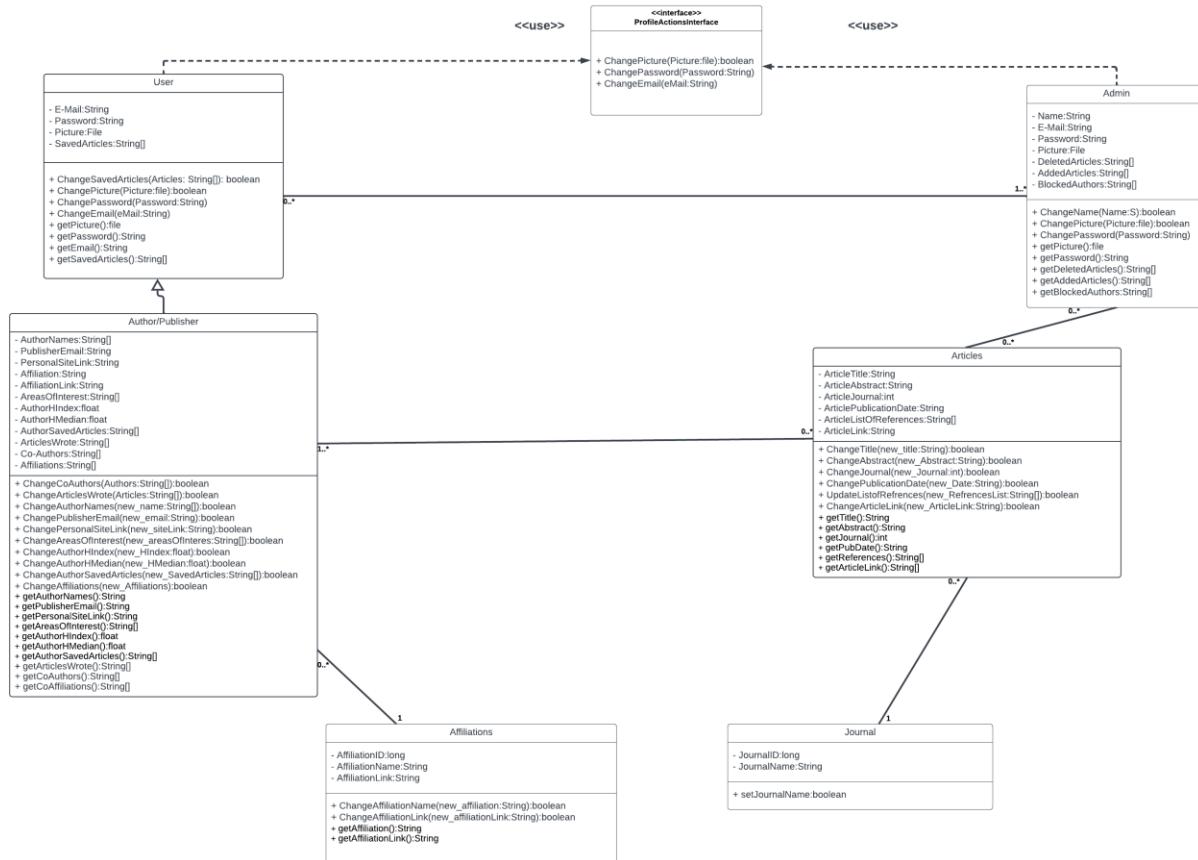


### 3.3.2. Class Diagram:

A class diagram is a graphical representation of the classes, interfaces, and associations in a software system. It provides a structural view of the system by showing the relationships between the classes and the attributes and methods they contain. For the KnowledgeVerse project, the class diagram would represent the different classes such as User, Publisher, Admin, Paper, Search, and others. The relationships between these classes and their attributes and methods would be depicted in the diagram. The class diagram would provide a high-level view of the system's architecture and would be useful for designing, documenting, and

communicating the system's structure to developers and stakeholders. It would also help in identifying potential design flaws or areas that require optimization.

*Figure 4: Class Diagram*



## 3.4. Process Flow:

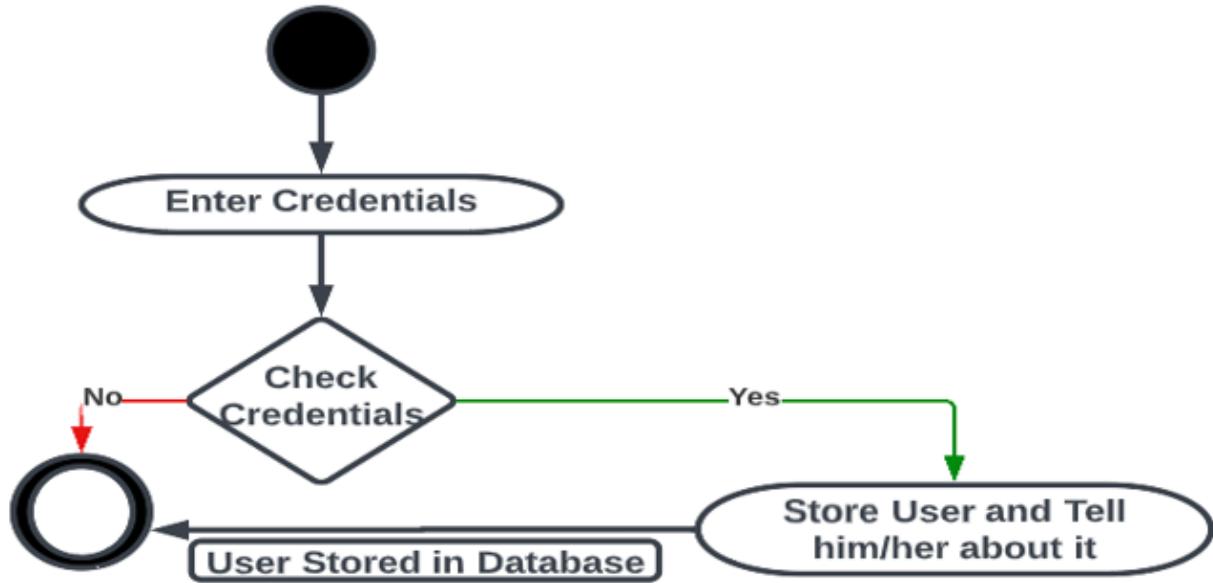
### 3.4.1. Activity Diagrams

The activity diagram of different activities which can be done in the knowledgeverse platform are shown below:

#### 3.4.1.1. User Sign-Up

The below diagram shows how the sign-up activity is being done in the platform.

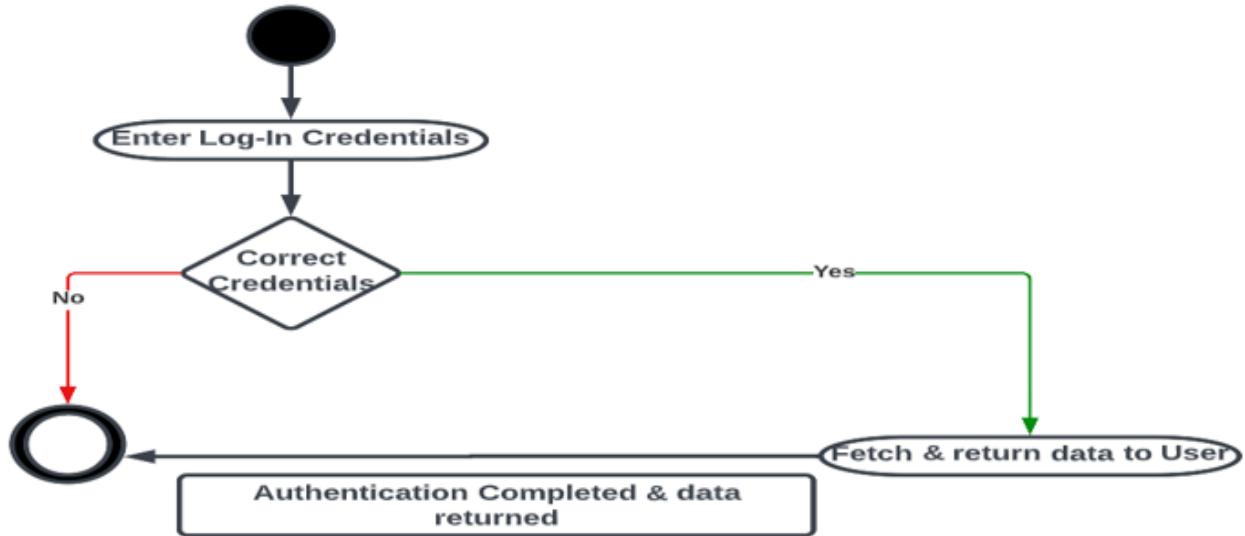
Figure 5: Activity Diagram - User Sign Up



#### 3.4.1.2. Activity Diagram of Log-In

The below diagram depicts how login activity is being done in the knowledgeverse platform.

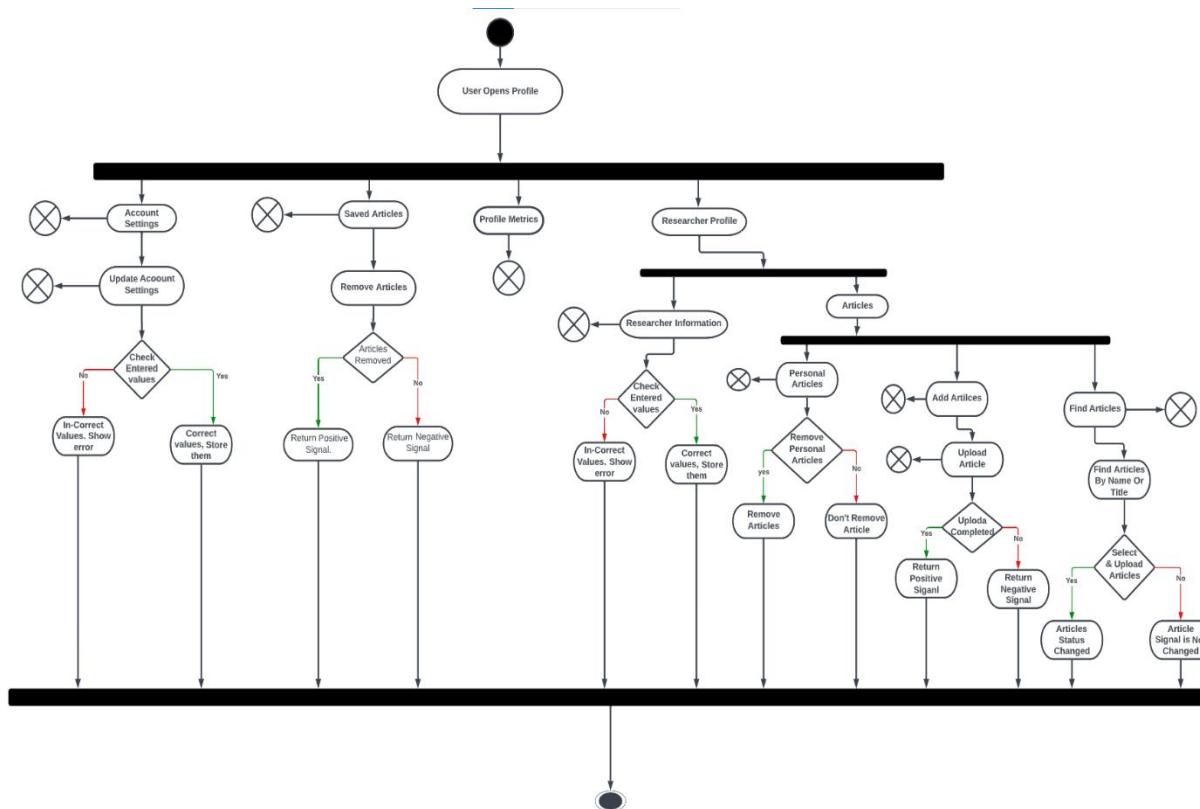
Figure 6: Activity Diagram -Log-In Activity Diagram



### 3.4.1.3. Publisher Activity Diagram

An activity diagram for the publisher role in KnowledgeVerse would illustrate the steps involved in publishing a new paper on the platform. This would include activities such as logging in to the system, uploading the paper.

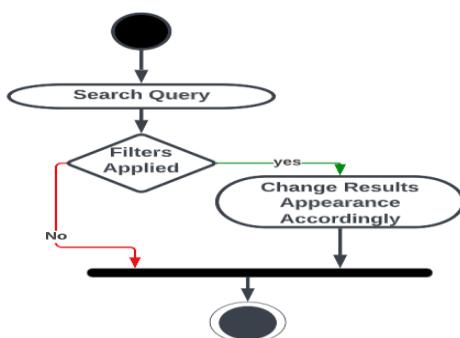
*Figure 7: Activity Diagram -Publisher/Author Activities*



### 3.4.1.4. User Search Activity Diagram

The diagram below shows how search is being done in the system.

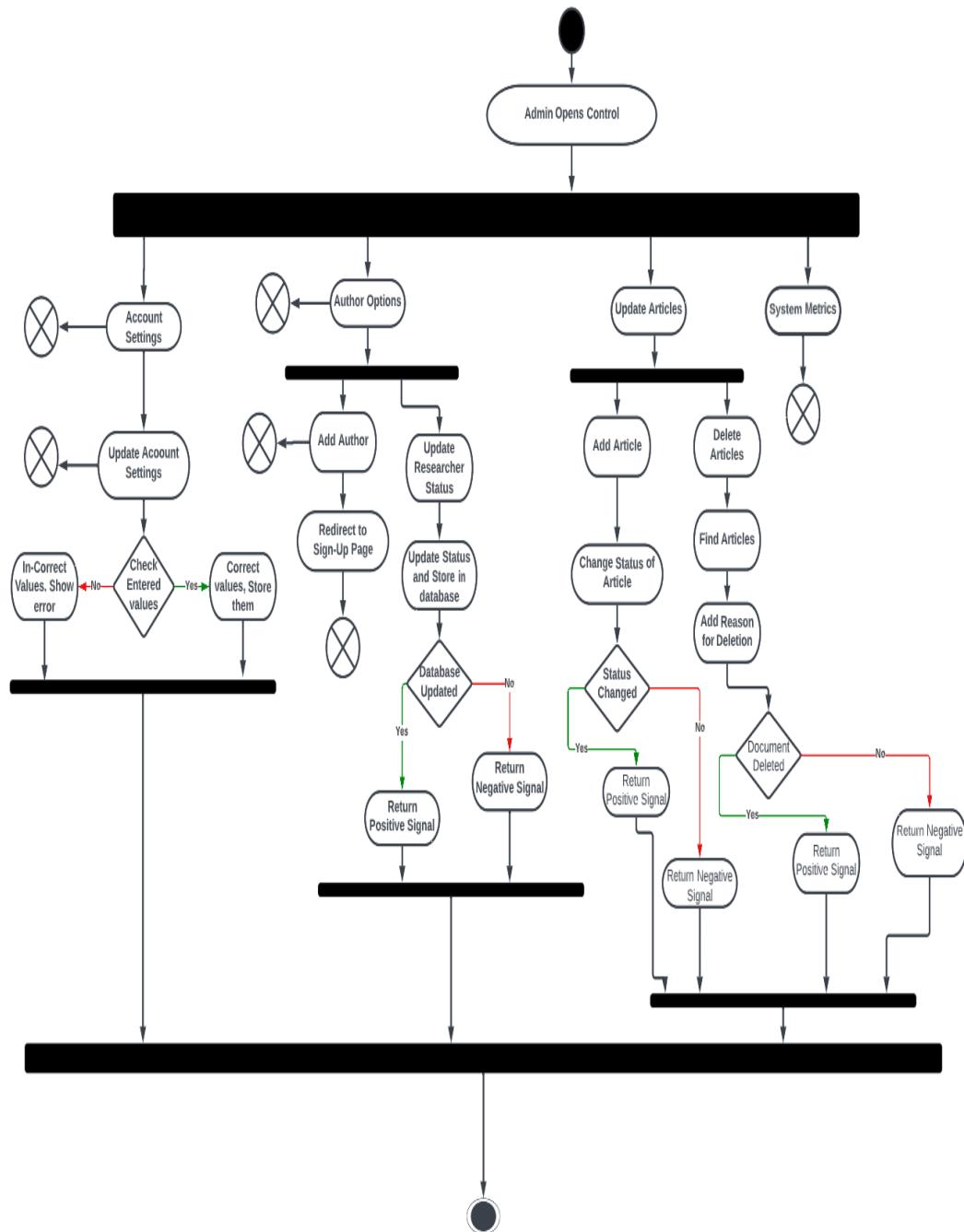
*Figure 8: Activity Diagram - User Search*



### 3.4.1.5. Admin Control

The different admin controls have been listed in this activity diagram below.

Figure 9: Activity Diagram -Admin



## 3.5. Design Models

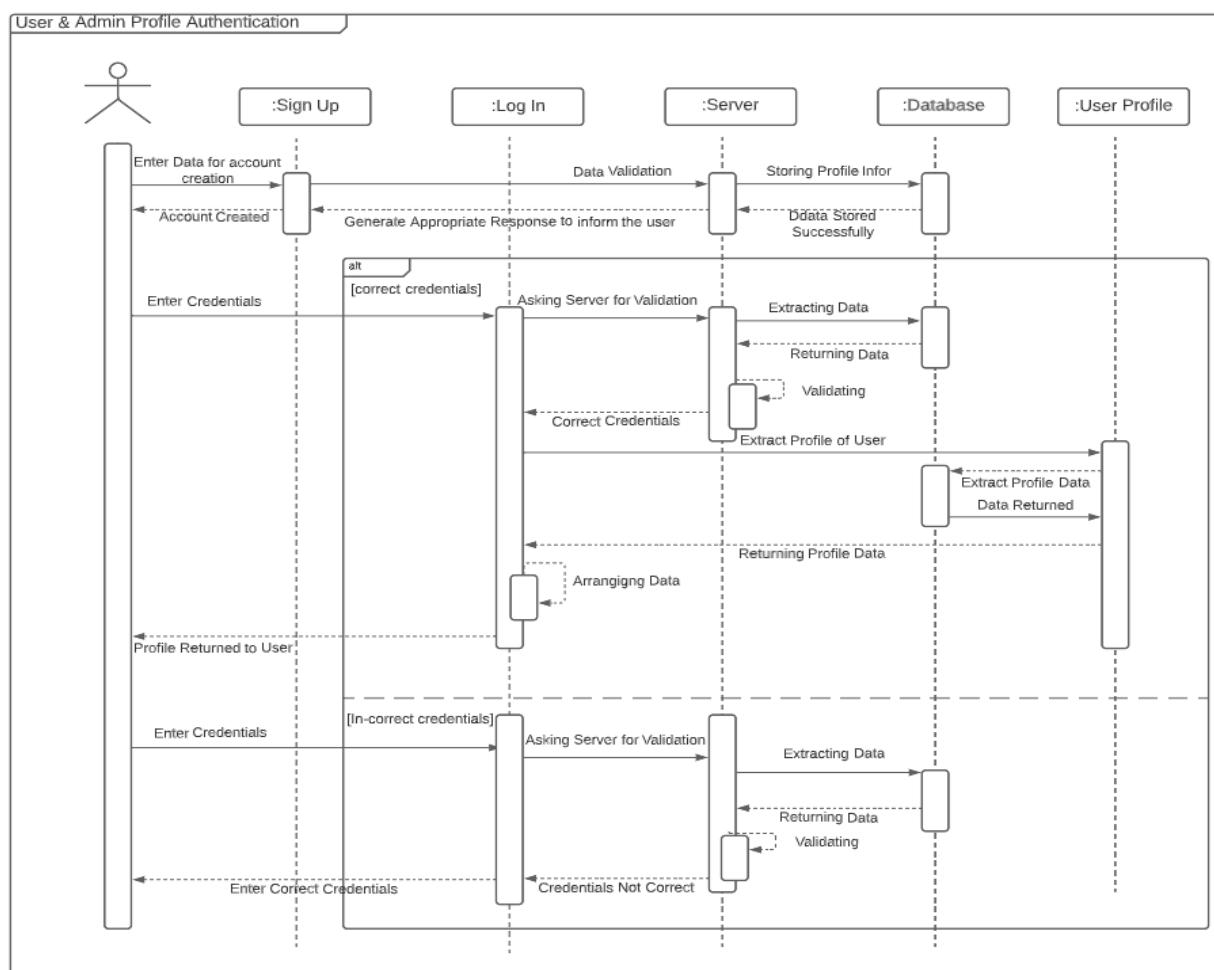
Design models are abstract representations of a system, process, or product that are used to aid in the design and development process. They provide a way to visualize and understand the structure and behavior of a system, and to communicate design decisions and requirements to stakeholders. Different design models are provided below.

### 3.5.1. Sequence Diagrams

Sequence diagrams are used to describe the interactions and messages exchanged between objects or components in a system. They provide a visual representation of the flow of control and the interactions between objects and are useful for capturing the dynamic behavior of a system.

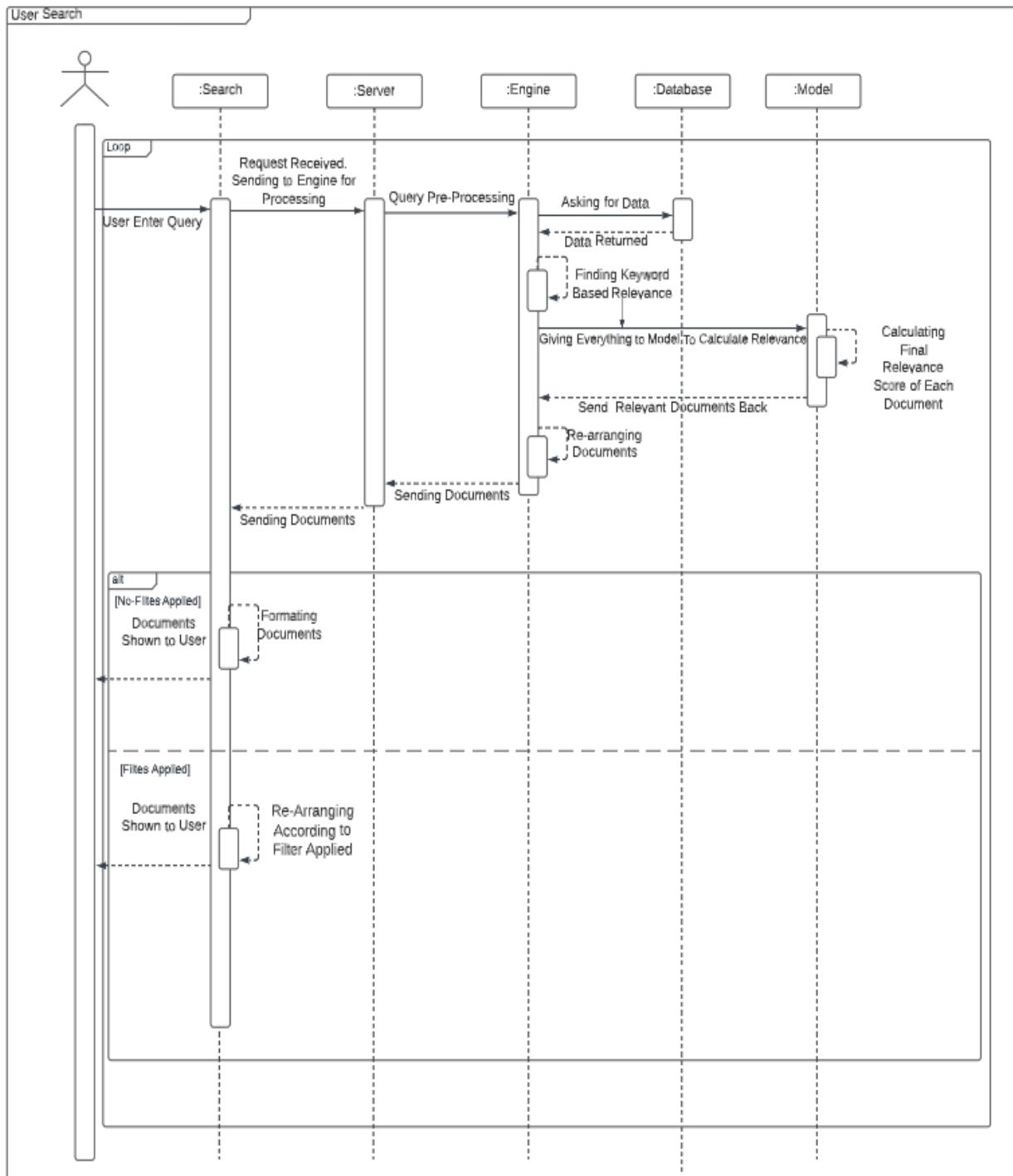
#### 3.5.1.1. Authentication Sequence Diagrams

Figure 10: Profile Authentication Sequence Diagram



### 3.5.1.2 User Search Sequence Diagrams

Figure 11: Sequence Diagram for User Search



### 3.5.1.3. Researcher Sequence Diagrams

Figure 12: Sequence Diagram Functionalities Provided to Researcher for updating its own profile.

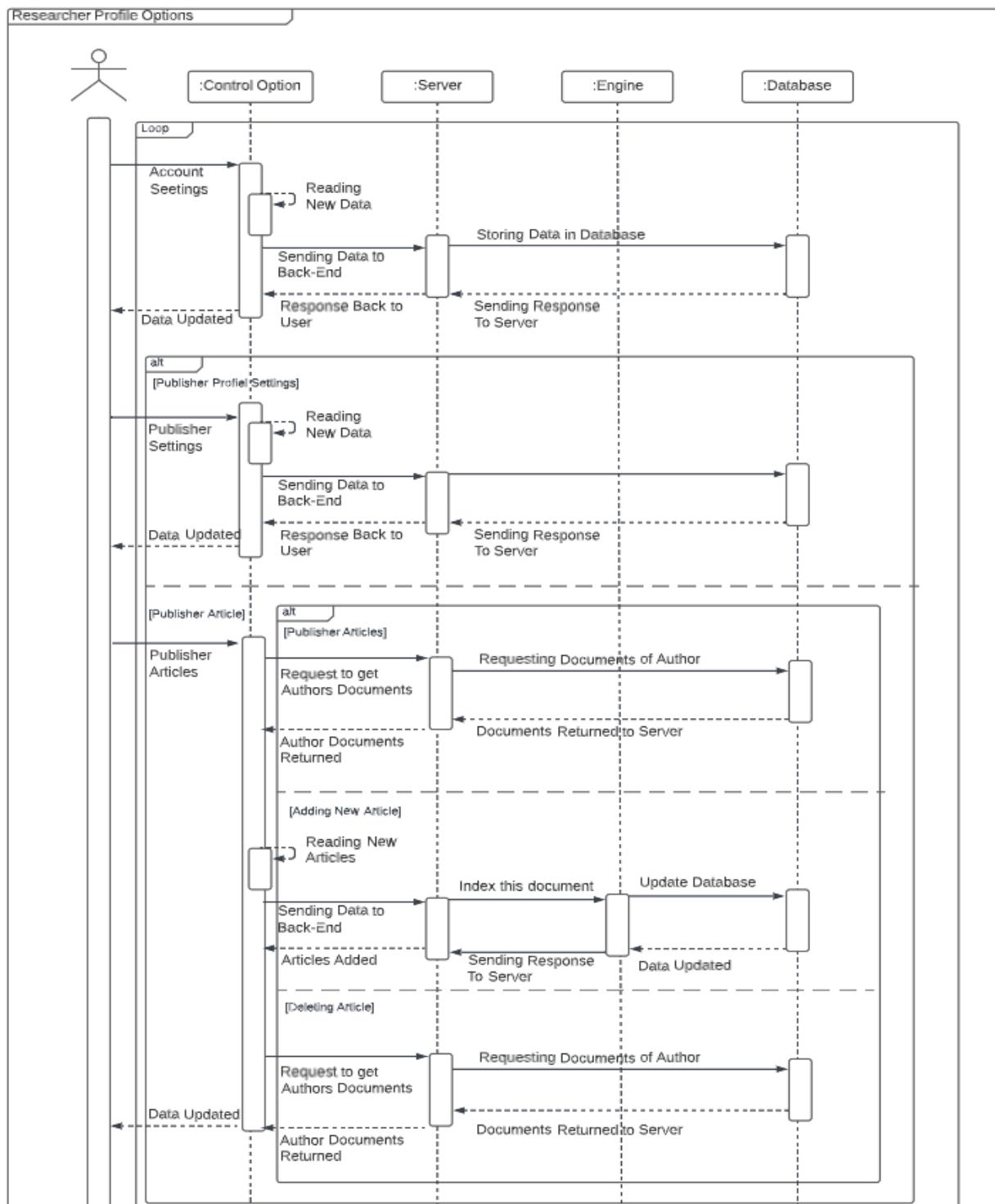
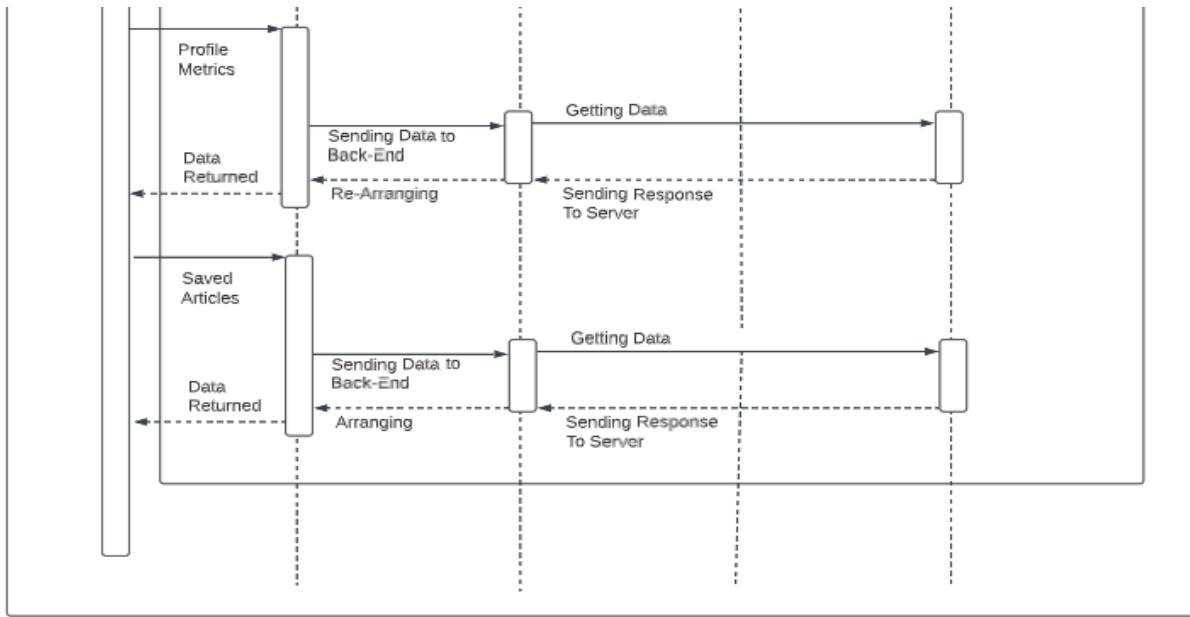
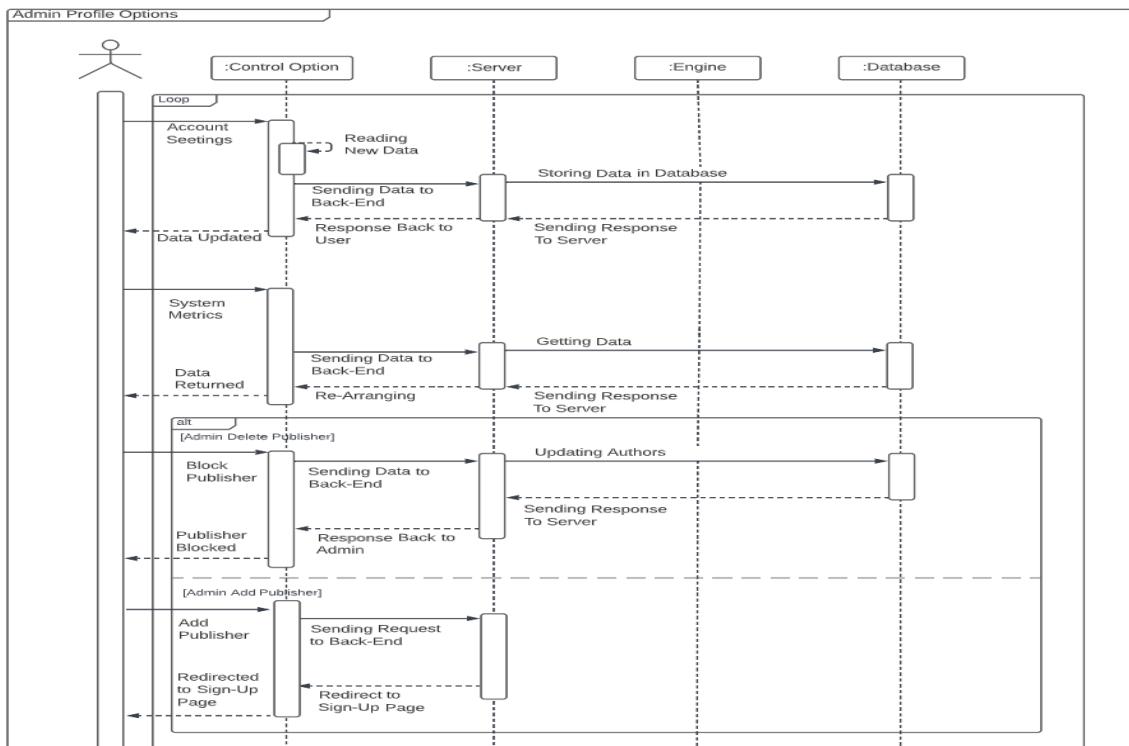


Figure 13: Figure-4 Carried

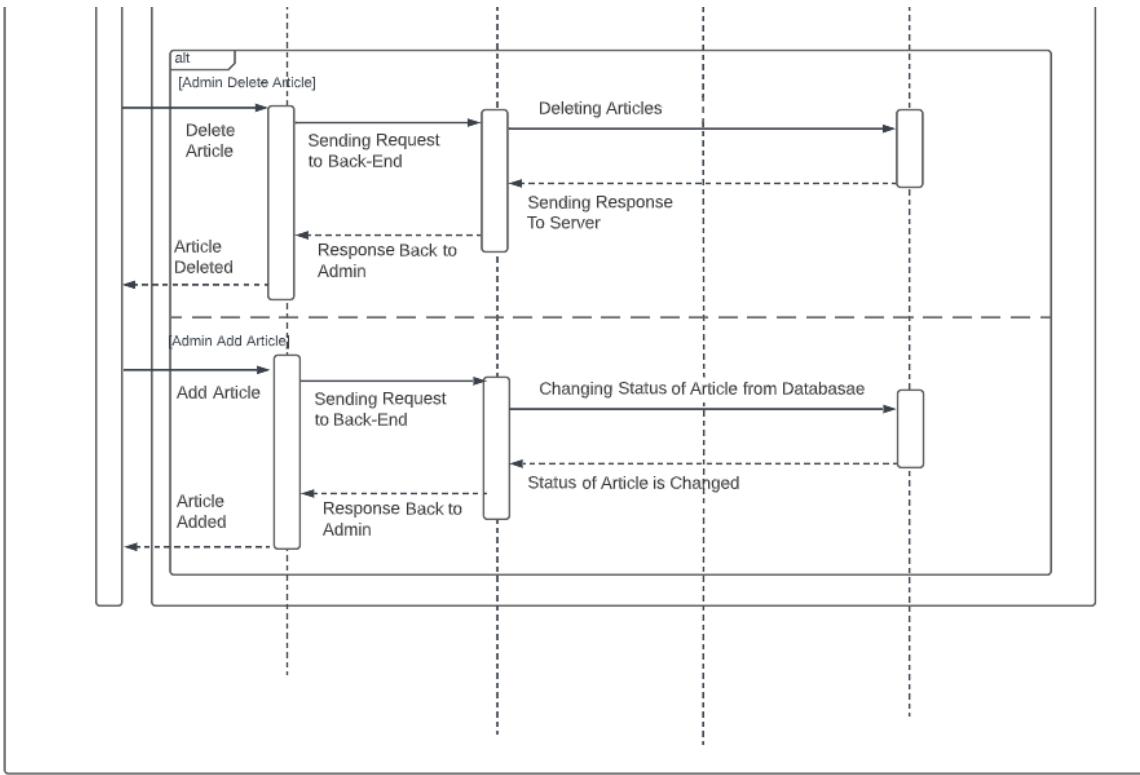


### 3.5.1.4. Admin Sequence Diagram

Figure 14: Sequence Diagram for Admin Functionalities



*Figure 15: Figure-6 Carried*



### 3.5.2. Data Dictionary

The different tables which have been used in the database schema have been shown below for a better understanding of the project.

### 3.5.2.1 User Table

*Table 2: Schema Table -User Table*

User Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
ID	INTEGER(10)	No	Yes	No		For identification of user
Email	VARCHAR(20)	No	No	No		User email
Password	VARCHAR(20)	No	No	No		User password

Picture	VARCHAR(20)	Yes	No	No		Picture address
Admin	BOOL	No	No	No	0	To check whether he is admin or not

### 3.5.2.2 Publisher Table

Table 3: Schema Table -Publisher Table

Publisher Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
ID	INTEGER(10)	No	Yes	No		For identification of publisher
Name	VARCHAR(20)	No	No	No		Publisher name
Email	VARCHAR(20)	No	No	No		Publisher email
Site	VARCHAR(20)	No	No	No		Publisher site
Affiliation Id	VARCHAR(20)	No	No	Yes		Affiliation id for publisher
AuthorHindex	Float(10)	No	No	No	0.0	Author performance measure
AuthorHmedian	Float(10)	No	No	No	0.0	Author performance measure
User Id	INTEGER(10)	No	No	Yes		User id for the Author

### 3.5.2.3 Admin Table

Table 4: Schema Table -Admin Table

Admin Table						
Name	Data Type	Nullable	PK	FK	Default	Comment

ID	INTEGER(10)	No	Yes	No		For identification of Admin
Name	VARCHAR(20)	No	No	No		Publisher name
User Id	INTEGER(10)	No	No	Yes		User id for the Author

### 3.5.2.4 Affiliations Table

Table 5: Schema Table - Affiliations Table

Affiliations Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
ID	INTEGER(10)	No	Yes	No		For identification of Affiliation
Name	VARCHAR(20)	No	No	No		Affiliation name
Link	VARCHAR(40)	No	No	No		Affiliation Url

### 3.5.2.5 Areas of Interest Table

Table 6: Schema Table -Areas of Interest Table

Areas of Interest Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Author ID	INTEGER(10)	No	No	Yes		For identification of Author
Interest	VARCHAR(20)	No	No	No		Interest topic

### 3.5.2.6 Author Names Table

Table 7: Schema Table -Author Names Table

Author Names Table						
Name	Data Type	Nullable	PK	FK	Default	Comment

Author ID	INTEGER(10)	No	No	Yes		For identification of Author
Name	VARCHAR(20)	No	No	No		Name

### 3.5.2.7 Blocked Authors Table

Table 8: Schema Table - Blocked Authors Table

Blocked Authors Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Author ID	INTEGER(10)	No	No	Yes		For identification of Author
Admin Id	VARCHAR(20)	No	No	Yes		For identification of admin who is blocking the author

### 3.5.2.8 Articles Table

Table 9: Schema Table - Articles Table

Article Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
ID	INTEGER(10)	No	Yes	No		Article unique id
Title	VARCHAR(20)	No	No	No		Title of the document
Abstract	VARCHAR(200)	No	No	No		Abstract of the document
Url	VARCHAR(20)	Yes	No	No		URL of the document
Author ID	INTEGER(10)	No	No	Yes		For identification of Author
Published Date	DATE	No	No	No		Date on which article is published
Journal Id	INTEGER(10)	No	No	Yes		Identifcation of

						journal
--	--	--	--	--	--	---------

### 3.5.2.9 References Table

Table 10: Schema Table -References Table

References Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Article ID	INTEGER(10)	No	No	Yes		For identification of current Article
Article ID_Referring	INTEGER(10)	No	No	Yes		Article being referred

### 3.5.2.10 CoAuthors Table

Table 11: Schema Table - CoAuthors Table

CoAuthors Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Article ID	INTEGER(10)	No	No	Yes		For identification of current Article
Author ID	INTEGER(10)	No	No	Yes		For identification of the document Author
Other Author ID	INTEGER(10)	No	No	Yes		For identification of other Author

### 3.5.2.11 Saved Articles Table

Table 12: Schema Table -Saved Articles Table

Saved Articles Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Article ID	INTEGER(10)	No	No	Yes		For identification of

						saved Article
User ID	INTEGER(10)	No	No	Yes		User who is saving the article

### 3.5.2.12 Articles Topic Table

Table 13: Schema Table - Article Topics Table

Article Topics Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Article ID	INTEGER(10)	No	No	Yes		For identification of saved Article
Topic ID	INTEGER(10)	No	No	Yes		Topic which is used in the article.

### 3.5.2.13 User Table

Table 14: Schema Table -Research Topics Table

Research Topics Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Topic ID	INTEGER(10)	No	Yes	No		Topic id
Topic Name	VARCHAR(20)	No	No	No		Topic name

### 3.5.2.14 Added Articles Table

Table 15: Schema Table - Added Articles Table

Added Articles Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Article ID	INTEGER(10)	No	No	Yes		For identification of

						added or approved Article
Admin ID	INTEGER(10)	No	No	Yes		Admin who has approved the article.

### 3.5.2.15 Deleted Articles Table

Table 16: Schema Table - Deleted Articles Table

Deleted Articles Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Article ID	INTEGER(10)	No	No	Yes		For identification of added or approved Article
Admin ID	INTEGER(10)	No	No	Yes		Admin who has approved the article.
Reason	VARCHAR(40)	Yes	No	No		Reason why are you deleting the article.

### 3.5.2.16 Journal Table

Table 17: Schema Table - Journal Table

Journal Table						
Name	Data Type	Nullable	PK	FK	Default	Comment
Journal ID	INTEGER(10)	No	Yes	No		Journal id
Journal Name	VARCHAR(20)	No	No	No		Journal name

# **Chapter 4**

## **4. System Testing**

System testing is an important phase of software development where the entire system is tested to ensure that it meets the specified requirements and performs as expected. For our project, KnowledgeVerse, system testing will involve testing the system as a whole, including the user interface, search algorithms, and database functionality. It will also involve testing different scenarios and use cases to ensure that the system works correctly and meets user expectations. Some important aspects of system testing for our project will include testing for performance, security, and usability, as well as ensuring that the system is reliable and robust. Proper system testing will help to identify and resolve any issues or bugs before the system is released to users, ensuring that our product meets the highest standards of quality and functionality. System testing generally includes two types of testing.

- Functional Testing
- Non-functional Testing

### **4.1. Functional testing:**

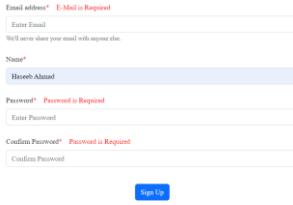
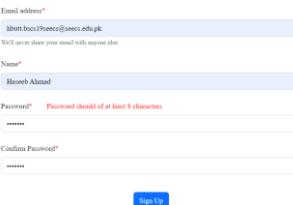
Functional testing is a type of testing performed on our project to validate if it meets the functional requirements specified in the requirements document. This testing involves validating each function and feature of the system against the specified requirements. The functional testing process includes identifying test scenarios, creating test cases, executing them, and analyzing the results. Functional testing can be performed manually or automated using testing tools. The goal of functional testing is to ensure that the system is working as expected, and all functions and features meet the user's requirements. The testing done for functional testing is mentioned below:

#### **4.1.1. Unit testing:**

#### 4.1.1.1 User Sign Up

The functional testing for user and publisher sign up, so that no one can sign up with the wrong data and our system can be avoided from getting corrupt or wrong data.

*Table 18: Test Case for User/Publisher Sign Up*

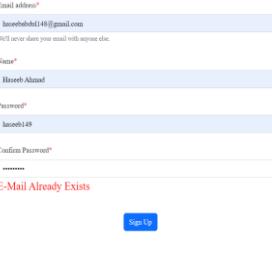
Index	Event	Attribute & Value	Expected Result	Result	OutPut
1	Both admin and User can create a new account for publisher to see its information. These sign-up credentials will be shared with publisher then the publisher can access all the resources information it wants.	Email: <u>hbutt.bsbs19seecs</u> @seecs.edu.pk Name: Haseeb Ahmad Password: haseeb149 Confirm Password: haseeb149	If no conflict occurs, User Account Created Successfully. User will be redirected to Log-In Screen.	Pass	 
2	Name, E-Mail, Password, Confirm Password Fields all are required.	Email: Missing Name: Haseeb Ahmad Password: Missing	User Account should not be created, and a prompt message should be given to fill the required fields.	Pass	
3	Password should be of at least 8 characters.	Email: Missing Name: Haseeb Ahmad Password: 1234567	Error message displayed on log in form. Failed to create user account.	Pass	

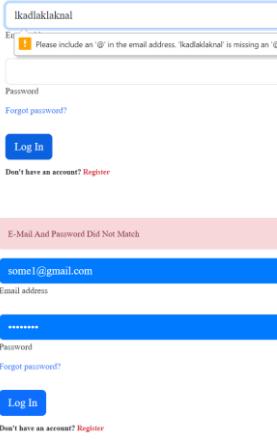
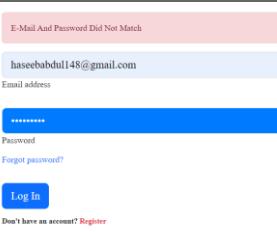
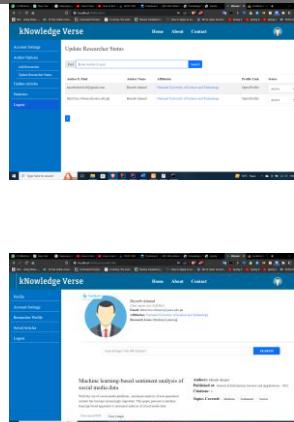
		Confirm Password: 1234567			
--	--	---------------------------------	--	--	--

#### 4.1.1.2. User Login:

The user or publisher can only login on entering the right credentials, else a proper message indicating the mistake will be shown to the user.

Table 19: Test case for user/publisher login

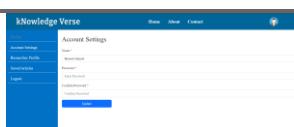
Index	Event	Attribute & Value	Expected Result	Result	Output
1	Both admin and User can create a new account for publisher to see its information. These sign-up credentials will be shared with publisher then the publisher can access all the resources information it wants.	Email: <u>hbutt.bscs19seec</u> <u>s@seecs.edu.pk</u> Name: Haseeb Ahmad Password: haseeb149 Confirm Password: haseeb149	If no conflict occurs, User Account Created Successfully. User will be redirected to Log-In Screen.	Pass	  
2	Authenticate user login after clicking on “Login” button with correct input.	Email: <u>hbutt.bscs19seec</u> <u>s@seecs.edu.pk</u> Password: haseeb149	Successfully logged in to the User Profile	Pass	

3	Authenticate user login after clicking on “Login” button with incorrect email input.	Email: <u>some1@gmail.co</u> m Password: aljbafakj	Error message displayed on log in form. Failed to log in to the User Profile	Pass	
4	Authenticate user login after clicking on “Login” button with incorrect email and password input.	Email: <u>haseebabdul148</u> <u>@gmail.com</u> Password: hello123	Error message displayed on log in form. Failed to log in to the User Profile	Pass	
5	After authenticating, redirect to proper Profile.	For Admin: Email: <u>haseebabdul148</u> <u>@gmail.com</u> Password: haseeb149  For User: Email: <u>hbutt.bscs19seec</u> <u>s@seecs.edu.pk</u> Password: haseeb149	If user Admin, it should be redirected to Admin Profile, otherwise redirected to User profile.	Pass	

#### 4.1.1.3. User Log-Out

When user/publisher presses the log out button, then the user/publisher should be redirected to the home page.

*Table 20: Test case for Log out*

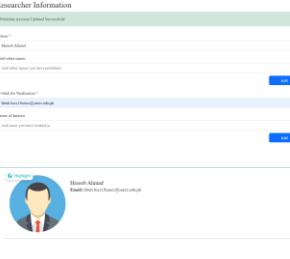
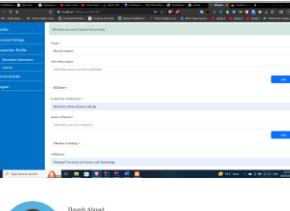
Index	Event	Attribute & Value	Expected Result	Result	Output
1	User Clicks on logout s/he should be logged out correctly and be redirected to home page of website.	Click on Log-out Button on Profile.	User Account Logged Out, completely.	Pass	 

#### 4.1.1.4. Publisher Profile:

The lot of publisher options will only be allowed when a user switch to the publisher account.

*Table 21: Test case for Publisher Options*

Index	Event	Attribute & Value	Expected Result	Result	Out Put
1	When user creates its publisher profile, only then s/he should be directed to	Name (Required), Other Published Names (Optional) Publisher Email (Required), Areas of Interests (Optional), Affiliation (Optional),	If Name and Publisher Email is not provided publisher profile should not be created.	Pass	

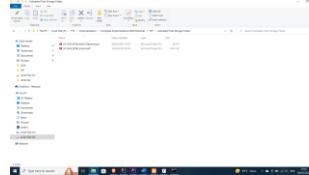
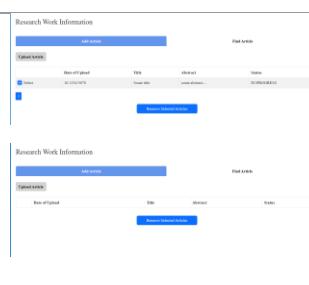
	access and add papers to their account.	Affiliation Home Page (Optional), Personal Site (Optional).			
2	User Enters the required fields only.	Name (Required), Publisher Email (Required)	If Name and Publisher Email is provided publisher profile should be created.	Pass	
3	User also enters the optional fields, then its author profile should be updated accordingly.	Other Published Names (Optional), Areas of Interests (Optional), Affiliation (Optional), Affiliation Home Page (Optional), Personal Site (Optional).	Optional fields should be updated.	Pass	

#### 4.1.1.5. Uploading Article for Indexing, from User Side:

These tests must be passed for smooth papers uploading.

Table 22: Test cases for Articles Uploading

Index	Event	Attribute & Value	Expected Result	Result	Out Put
1	Publisher should be allowed to upload document to our platform for	DOI (Required), Title (Required), Abstract (Required), Year (Required), Journal Name (Optional),	If any of the required field is missing, paper should not be submitted to web portal.	Pass	

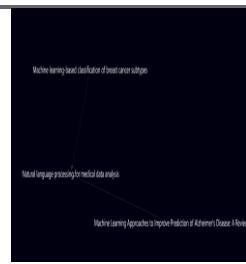
	indexing.	Paper Author Names (Required), PDF File (Required)			
2	User Enters the required fields only.	Paper will be uploaded with ‘kNowledgeVerse’ journal name.	If Journal Name is set to empty paper should be accepted and uploaded on directory.	Pass	
3	All Uploaded articles of a specific publisher should be displayed to him or her, with status of paper being displayed on our website.		Publisher should be able to see all uploaded articles, with their respective status.	Pass	
4	Publisher should be able to delete its paper from our platform.	Required Paper should be selected before proceeding.	Selected list of papers should be deleted from user Account.	Pass	

5	Publisher should be able to search and find papers from already indexed documents and should be able to be added them to their profile.	Required Paper should be selected before proceeding.	Selected list of papers should be deleted from user Account.	Pass	 
---	---	--	--	------	--

#### 4.1.1.6. 3D Graph Display:

The citation graph of every paper should be shown on clicking the citation graph button.

*Table 23: Test case for Citation Graph Display*

Index	Event	Expected Result	Result	Output
1	When a user chooses to see the Citation Graph of Paper. 3D Citation Graph of paper should be displayed. Set	3D Graph of Citation Network should be displayed.	Pass	

#### 4.1.1.7. Save Articles:

Users can add or remove article from the saved articles side.

*Table 24: Test case for save articles*

Index	Event	Attribute & Value	Expected Value	Result

1	User should be able to save the papers which s/he wants to view or read later.	User must be logged in.	Paper should be added to the user saved list when user clicks on save article button.	Pass
2	User should be able to delete the saved papers which s/he wants to delete from saved list	User must be logged in.	Paper should be deleted from the user saved list when user clicks on delete article button.	Pass

#### 4.1.1.8. User Search

Users should be able to search any keyword or query on the platform.

*Table 25: Test case for Query search*

Index	Event	Attribute & Value	Expected Result	Result
1	User should be able to search on any keyword and can see results relevant to that query.	Query:"NLP"	Articles or papers relevant to NLP should be shown to the users.	Pass

2	If the papers are not a little relevant to the user query then, it shouldn't be shown to the users.	Query: "Medical sciences"	If our dataset don't have papers on medical sciences then no result should be shown in the result field.	Pass
---	---	------------------------------	--	------

#### 4.1.1.9. Filtered Search(*Date, citations, in between date, journal*)

Users should be able to see the results based on different filters.

*Table 26: Test case for filtered search*

Index	Event	Attribute & Value	Expected Result	Result
1	User can see the result on the basis of citation score filter	Citation score filter selected	The paper with the most citation should be shown on the top	Pass
2	User can see the results on the basis of journal.	Journal selected:"IEEE"	The relevant papers which are being published on IEEE should be shown on the results field.	Pass

3	User can see the latest articles or papers on the top.	Sort by latest is selected	The relevant papers which are being published latest should be shown on the top.	Pass
4	User can see the results between some two dates	Two dates are being given: From: 2000 To: 2020	The relevant papers which are being published between these dates should be shown	Pass

#### 4.1.1.10. Related Articles:

The related articles will be shown when user selects a single paper.

*Table 27: Test case for Related Articles*

Index	Event	Attribute & Value	Expected Result	Result
1	Related articles should be shown when user selects an article from the search field.	Paper selected from the search field	The related paper from the selected paper should be shown	Pass

#### *4.1.1.11. Cited Articles:*

The papers which are citing this paper should be shown on clicking the citation button.

*Table 28: Test case for cited articles*

Index	Event	Attribute & Value	Expected Result	Result
1	Citations will be shown on clicking the citation button	Click on citation button	The papers which are citing this paper should be shown.	Pass

#### *4.1.1.12. Publisher Profile:*

Users can see the publisher profile of different publishers.

*Table 29: Test case for publisher profile*

Index	Event	Attribute & Value	Expected Result	Result
1	Users can see the profile of different publishers.	User clicked on any Auhtor's name.	The publisher profile will be opened in a specific format in which his papers, and contributions can be seen.	Pass

#### *4.1.1.13. Admin Stats:*

Admin can view the stats of users and publishers.

*Table 30: Test case for Admin stats*

Index	Event	Attribute & Value	Expected Result	Result
1	Admin should be able to view the stats of users and publishers	Logined using Admin credentials	The user's and publishers' stats in the form of graph wil be shown	Pass

#### *4.1.1.14. Managing Authors*

Admin can manage the authors and their different details.

*Table 31: Test case for Managing Authors*

FR-7	Event	Attribute & Value	Expected Result	Result	Output
1	Admin Should be able to see all the authors to manage them.		All Publishers should be displayed to Admin	Pass	
2	Admin Should be able to find specific publisher, using its email	Publisher Email (Required)	If required Email, is provided, then find the publisher with that email.	Pass	
3	Admin should be able to find and see all the	Publisher Email (Should be empty)	Publisher Email Field should be empty.	Pass	

	publishers after searching for specific publisher.				
4	Admin Should be able to block and active any publisher.	Blocked Status field should be selected.	Publisher Status should be changed accordingly.	Pass	
5	Admin should be able to unblock any blocked publisher.	Active Status field should be selected.	Publisher Status should be changed accordingly.	Pass	

#### 4.1.1.15.. Managing Articles:

User can add or remove article from the saved articles side.

*Table 32: Test case for managing articles*

Index	Event	Attribute & Value	Expected Value	Result
1	Admin can accept or reject newly uploaded articles from publishers	Papers uploaded from publisher site	Admin accepted or rejected from admin must remain in the database and only gets	Pass

			indexed if its status is “accepted”	
2	Admin can reject the already accepted paper from the indexed list	Paper must be indexed	Paper should be removed from the indexed list.	Pass

#### 4.1.2. Integration Testing:

Integration testing is a crucial aspect of software testing, especially for complex systems such as KnowledgeVerse. It is the process of testing how different modules of the system work together to ensure that they function correctly as a whole. In the case of KnowledgeVerse, integration testing would involve testing the different components of the system, including the search engine, database, user interface, and various algorithms, to ensure that they work together seamlessly. This testing would ensure that the system is functioning optimally and that the various components are communicating with each other correctly. Integration testing is an important step in ensuring that KnowledgeVerse is a reliable and high-performing platform for users.

The objective of integration testing is to ensure that every module is working perfectly fine together. So, the tests and their results for integration testing are shown below.

##### 4.1.2.1 User/Publisher Testing:

Index	Event	Expected Value	Result

1	User can make a user account on our platform	User signed up on knowledgeverse with the right credentials required for it	Pass
2	User signed in to knowledgeverse platform	User signed in to knowledgeverse with the right credentials and can access different utilities	Pass
3	User can search different queries on knowledgeverse and can see results and also apply filters on them	User searched some query and got results on which s/he applied filters and got modified results.	Pass
4	User can see the related papers after clicking on a paper	Related papers are being shown after selecting a single paper	Pass
5	User can see the citation of a paper when clicked the citation button	The papers which are citing a particular paper will be shown	Pass
6	User can search a category directly after clicking on category button from the home page	The user can search a category directly from the home page and that categorized papers will be shown on the search page.	Pass

#### 4.1.2.2 Admin Testing:

Index	Event	Expected Value	Result
1	Admin can add or reject papers from the manage papers	The paper accepted will be added to the indexed papers while	Pass

		rejected papers will be removed from the indexed list	
2	Admin can view stats of the users and publishers.	The users and publishers stats will be shown in the form of graph to the admin.	Pass

## 4.2. Non-functional Testing:

Nonfunctional testing is a type of testing that focuses on the performance, reliability, scalability, and security of a software system. It is concerned with evaluating the system's behavior under different conditions and verifying that it meets the nonfunctional requirements specified for the project.

For the Knowledgeverse project, nonfunctional testing would involve evaluating the system's ability to handle large volumes of data, its response time, the accuracy and completeness of search results, its availability and reliability, and its security features such as authentication, authorization, and encryption.

### 4.2.1. Performance Testing:

Performance testing is a type of non-functional testing that evaluates the speed, stability, scalability, and responsiveness of a system under a specific workload. For the Knowledgeverse project, performance testing is important to ensure that the system can handle a large number of concurrent users and process a large volume of data efficiently.

Metric	Meaning	Value
First contentful load	The first div or container appeared on the screen.	<6ms
Speed Index	How quickly the whole page is	2sec

	being populated on the screen	
Time to interactive	The amount of time it takes to be fully operational	3sec
Request Handling	How much requests it can handle concurrently	200+
Browser Independent	It can run on any browser	Pass

#### 4.2.2. Security Testing

Security testing is an essential part of the KnowledgeVerse project, which aims to ensure that the system is secure and protected from potential threats. Security testing involves identifying and addressing vulnerabilities and weaknesses in the system to prevent unauthorized access, data breaches, and other security risks.

Test Case	Status
Only verified users are allowed to login	Pass
Forms are only submitted on properly filling	Pass
User cannot access/edit publisher and admin data	Pass
Publisher cannot access admin stats	Pass

# **Chapter 5**

## **5. System Integration and Deployment:**

### **5.1. System Integration**

System Integration is the process of combining different subsystems or components into a single system that functions as one entity. It involves connecting different systems, software, and hardware together to ensure that they work efficiently and effectively. In the context of the KnowledgeVerse project, system integration involves combining various modules of the system such as machine learning models, APIs, and front-end modules into a single system.

Deployment, on the other hand, is the process of making the system available for use by end-users. In the case of KnowledgeVerse, the deployment process involves making the entire system available on a server or a cloud platform. This involves configuring the server, setting up databases, installing required software, and deploying the code.

For KnowledgeVerse, the deployment process is crucial, as it involves deploying the machine learning models, APIs, and front-end modules separately. This ensures that the system is scalable and can handle increased traffic and load. Furthermore, it makes it easier to maintain and update the system as different components can be updated or replaced without affecting the entire system. Overall, system integration and deployment are critical processes in the development and deployment of any software system, including KnowledgeVerse. By ensuring that all the different modules work together efficiently, and by deploying the system correctly, we can ensure that the system is reliable, scalable, and easy to maintain.

The integration of models with the database is done such that the papers are first get from the indexed papers list from the category model and then is passed to the relevance model which calculates the relevance of the papers with the query and then freshness score and citation score is being integrated into the final score and then final results are being shown to the users. This is how search related things are being integrated. While the sign in and sign up are being integrated and managed using routes and states. Only entering the right credentials for user

sign in, the user automatically gets moved to the home page where he can search queries and see different authors profiles and papers.

#### 5.1.1. Email Integration:

Email integration is an important aspect of any web-based platform that requires communication between users and the system administrator. In Knowledgeverse, we have integrated email functionality using the Spring Boot Starter Mail library. This integration allows us to send the submitted contact us form details directly to the administrator's email address.

With this email integration, the system administrator can quickly access the contact us forms submitted by the users, review them, and respond promptly. This can help in addressing the users' concerns and queries more efficiently, leading to a better user experience.

The integration has been implemented using the popular Spring Boot framework, which provides a robust and flexible platform for building web applications. The Spring Boot Starter Mail library provides a simple way to send emails using SMTP or other email protocols.

Overall, email integration is an essential part of Knowledgeverse's system integration and deployment. It helps in streamlining the communication between the users and the system administrator, leading to a more efficient and responsive platform.

## 5.2. System Deployment:

System deployment for Knowledgeverse involves deploying multiple components separately. Solr has already been deployed on AWS, while the machine learning models, backend and frontend are yet to be deployed. The machine learning models will be deployed separately, with each model running as a service. The backend, which consists of APIs, will also be deployed separately. This will ensure that each component can scale independently, making the system more resilient and fault tolerant.

For deploying the frontend, we will use a cloud platform such as AWS or Heroku. The frontend can be deployed using a content delivery network (CDN) or serverless architecture for better scalability. Overall, the deployment process for Knowledgeverse will involve multiple steps, with each component being deployed separately to ensure maximum efficiency and scalability.

# **Chapter 6**

## **6. Key User Interfaces**

This chapter includes all the images of the web interface of the knowledgeverse platform. We had added every important interface of website along with the description in this chapter below:

### **Home Page:**

The home page is the first point of entry for anyone visiting the Knowledgeverse platform. It serves as a gateway to all the knowledge and information available on the platform. The home page has been designed to be intuitive and user-friendly, ensuring that users can easily find what they are looking for.

The search bar, located prominently at the top of the page, is the first thing that users will notice. Users can enter a keyword or a phrase to search for relevant content on the platform. Alternatively, they can use the categories section located just below the search bar. The categories section is based on the most commonly searched topics on the Knowledgeverse platform, and users can select a category to refine their search.

Directly below the categories section is a list of the most viewed or cited papers on the platform. This section highlights the most popular and widely read content on the platform and provides users with a quick and easy way to access this information.

Further down the page, there is a brief description of the platform. This section provides users with an overview of the Knowledgeverse platform, its purpose, and the value it offers. It aims to convey the mission of the platform and the benefits of using it.

Finally, the home page ends with the footer, which provides users with links to important information about the platform, such as the terms and conditions, privacy policy, and contact information. The footer also includes links to social media accounts, which users can follow to stay up to date with the latest developments on the platform.

Overall, the home page of the Knowledgeverse platform is designed to be simple, intuitive, and user-friendly. It provides users with a range of options for accessing the information they need, whether it be through a direct search or by browsing through popular categories. The platform's user-friendly design, combined with its rich and diverse content, makes it an ideal destination for anyone seeking knowledge and information. The home page interface is shown below:

*Figure 16: User Interface: Home Page*

The screenshot displays the Knowledgeverse platform's home page with a dark blue header and footer. The header features a navigation bar with 'Home', 'About', and 'Contact' links, a user profile icon, and a search bar with placeholder text 'Search for a paper, author, or topic'. Below the header, the title 'kNowledge Verse' is displayed, followed by the subtitle 'A free, AI-powered research tool for scientific literature'. A search bar with a magnifying glass icon is positioned below the title.

**Top Research Topics:** MODEL, ET, TRANSLATION, TEXT, DATA, AL, APPROACH, LANGUAGE, DOCUMENT, SPEECH

**Recent Papers:**

- SENTENCE**: Title Generation with Quasi-Synchronous Grammars. A brief summary discusses the process of condensing a source text into a shorter version while preserving its finer-grained content. It notes that humans summarize on a daily basis efficiently, yet the automatic production of high-quality summaries remains a challenge. [See More](#)
- TYPE**: A Simple Domain-Independent Probabilistic Approach to Generation. This paper focuses on the problem of generating descriptive text for a document represented by a set of database records. It highlights that existing generation systems can be engineered to obtain good performance on particular domains (e.g., Dale et al.). [See More](#)
- RECORD**: A Simple Domain-Independent Probabilistic Approach to Generation. This paper addresses the problem of generating descriptive text given a web page represented by a set of database records. It notes that existing generation systems can be engineered to obtain good performance on particular domains (e.g., Dale et al.). [See More](#)

**Most Cited Papers:**

- CITATIONS**: Summarizing Contrastive Viewpoints in Opinionated Text. A brief summary discusses the growing popularity of opinionated text and the need for tools that can summarize opinions expressed in such text so that a user can easily digest them. It presents a study on how to summarize opinionated text. [See More](#)
- TEXT**: Automatically Producing Plot Unit Representations for Narrative Text. A brief summary discusses the challenges of representing narrative stories and generating summaries. Plot units are fundamentally different from story representations that preceded them because they focus on the plot. [See More](#)
- CITATIONS**: Inducing Word Senses to Improve Web Search Result Clustering. A brief summary discusses the increasing amount of text available online and the need for better search engines. It presents a study on how to induce word senses to improve web search results. [See More](#)

**A New Beginning**

Our vision is to create an ecosystem that fosters collaboration, innovation, and excellence in research. With this new beginning, we are committed to developing cutting-edge technologies that will enhance the user experience, streamline workflows, and enable more effective knowledge discovery. We believe that this new chapter will usher in a new era of scholarly excellence and transform the way we approach research and scholarly communication.

**Purpose**  
The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that lord somehow crossed over into an entirely new dimension.

**Categories**  
MODEL, ET, TRANSLATION, TEXT, DATA, AL

**Send Your Suggestions**  
Suggestion Title  
Enter your suggestions here

Copyright © 2023 All Rights Reserved

[Facebook](#) [Twitter](#) [Instagram](#) [LinkedIn](#)

## **Search Page:**

The search page of the Knowledgeverse platform provides users with a seamless experience to explore and access relevant research papers. The page starts with a header that displays the platform logo and menu options to navigate to other pages. The search bar, located at the top of the page, is the primary tool for users to search for papers by typing in a keyword or phrase.

To further enhance the search experience, the platform provides a filter option located just below the search bar. The filter allows users to apply specific search criteria to narrow down their results. Users can apply filters based on the date of publication, citation score, author, and other criteria, enabling them to find the most relevant papers that meet their specific needs.

After applying the filters, the page displays a list of research papers relevant to the user's query. The list provides a brief description of each paper, including the title, authors, abstract, and other important details. Users can quickly browse through this list to find the most relevant papers.

The platform also ensures that the papers listed are sorted according to relevance, making it easier for users to find the most relevant papers quickly. Additionally, users can click on any paper to view more details about it, including the full text, author details, citations, and other important information.

Figure 17: User Interface - Search Page

The screenshot shows the kNowledge Verse search interface. At the top, there's a navigation bar with links for Home, About, Contact, and Login. Below the navigation is a search bar with the placeholder "natural language processing". Underneath the search bar are filters for "Select Date" (set to 2009), "Sort By Date" (unchecked), "Sort By Citations" (unchecked), and a dropdown menu for "More". A search button is located to the right of the search bar.

The main content area displays a list of research papers:

- Better Punctuation Prediction with Dynamic Conditional Random Fields**  
Outputs of standard automatic speech recognition (ASR) systems typically consist of utterances where important linguistic and structural information (e.g., stress areas, sentence boundaries, punctuation symbols, etc.) is not available. Such information is often lost due to the nature of the ASR system or the post-processing stage.
- Joint Training and Decoding Using Virtual Nodes for Cascaded Segmentation and Tagging Tasks**  
There is a typical class of sequence labeling tasks in many natural language processing (NLP) applications, which require some a cascade of segmentation and tagging subtasks. For example, many Asian languages work in Japanese and Chinese while others
- Modeling Organization in Student Essays**  
Automatic essay scoring, the task of employing computer technology to evaluate and score written text, is one of the most important educational applications of natural language processing (NLP) (see Slemmons and Barzilay (2003) and Slemmons et al. (2010) f
- Automatic Keyphrase Extraction via Topic Decomposition**  
Keyphrases are defined as a set of terms in a document that give a brief summary of its content for readers. Automatic keyphrase extraction is widely used in information retrieval and digital library (Turmy, 2000; Nguyen and Kao, 2007). Keyphrase extraction is
- NLP on Spoken Documents Without ASR**  
Can we do IR-like tasks without ASR? Information retrieval (IR) typically makes use of simple features that count terms within across documents such as term frequency (TF) and inverse document frequency (IDF). Currently, to compute these features, it is n
- A Game-Theoretic Approach to Generating Spatial Descriptions**  
Language is about successful communication between a speaker and a listener. For example, if the goal is to reference the target object O1 in Figure 1, a speaker might choose one of the following two utterances: (a) eight of O2 (b) on O3 Although both
- Storing the Web in Memory: Space Efficient Language Models with Constant Time Retrieval**  
We present three novel methods of compactly storing very large n-gram language models. These methods use substantially less space than all known approaches and allow n-gram probabilities or counts to be retrieved in constant time, at speeds competitive
- Improving Translation via Targeted Paraphrasing**  
For most of the world's languages, the availability of translation is limited to two possibilities: high-quality high cost, via professional translators, and low quality at low cost, via machine translation (MT). The spectrum between these two extremes
- Facilitating Translation Using Source Language Paraphrase Lattices**  
In recent years, statistical MT systems have been easy to develop due to the rapid explosion in data availability, especially parallel data. However, in reality there are still many language pairs which lack parallel data, such as Urdu/English, Chinese/I
- Exploiting Conversation Structure in Unsupervised Topic Segmentation for Emails**  
With the ever increasing popularity of emails and web technologies, it is very common for people to discuss issues, events, agenda or tasks by email. Effective processing of the email contents can be of great strategic value. In this paper, we study the

Each paper entry includes a "Download PDF" button, a "Save Article" button, and a "View Graph" button. To the right of the list, there are detailed author information, publication details, citation counts, and topic coverage for each paper. At the bottom of the page, there are footer links for Purpose, Categories, and Send Your Suggestions, along with social media sharing icons.

## Single Paper Page:

After selecting a paper from the search page or home page, you will be navigated to the single paper page where you will see the specific details of a paper. Here the complete abstract, citation details and related or recommended papers will be shown. The user can see the Author list and the categories in which the paper falls. The interface of Single Paper page is as follows:

**Figure 18: User Interface - Single Paper**

### Title Generation with Quasi-Synchronous Grammar

Published on: 2010

Summarization is the process of condensing a source text into a shorter version while preserving its information content. Humans summarize on a daily basis and effortlessly, yet the automatic production of high-quality summaries remains a challenge. Most work today focuses on extractive summarization, where a summary is created by identifying and subsequently concatenating the most important sentences in a document. The advantage of this approach is that it does not require a great deal of linguistic analysis to generate grammatical sentences, assuming the source document was well written. Unfortunately, extracts generated this way are often documents of low readability and text quality, and contain much redundant information. The conciseness can be improved when sentence extraction is interleaved with sentence compression, where words and clauses are deleted based on rules typically operating over parsed input (Jing, 2000; Daume? III and Marcu, 2002; Lin, 2003; Daume? III, 2006; Zajic et al., 2007; Martins and Smith, 2009). An alternative abstractive or “bottom-up” approach involves identifying high-interest words and phrases in the source text, and combining them into new sentences guided by a language model (Bansko et al., 2000; Soricean and Marcu, 2007). This approach has the potential to work well, breaking out of the single-sentence paradigm. Unfortunately, the resulting summaries are not always coherent? individual constituent phrases are often combined without any semantic constraints? or grammatical beyond the a priori horizon imposed by the language model. Constituent deletion and recombination are merely two of the many rewrite operations possible: socal editors and abstractors employ when creating summaries (Jing, 2002). Additional operations include truncating sentences, aggregating them, and paraphrasing at word or syntax level. Furthermore, professionals write summaries in a task-specific style. News headlines for example are typically short (three to six words), written in the present tense and active voice, and often leave out forms of the verb be. There are also different ways of writing a headline either directly by stating what the document is about or indirectly by raising a question in the reader’s mind, which the document answers. The automatic generation of summaries similar to those produced by human abstractors is challenging because of the many constraints imposed by the task: the summary must be maximally informative and minimally redundant, grammatical, coherent, adhere to a pre-specified length and stylistic conventions. Importantly, these constraints are conflicting: the deletion of certain phrases may avoid redundancy but result in ungrammatical output and information loss. In this paper we propose a model for summarization that attempts to capture and optimize these constraints jointly. We learn both how to select the most important information (the content), and how to render it appropriately (the style). Selection preferences are learned discriminatively, while a quasi-synchronous grammar (QG, Smith and Eisner 2006) captures rendering preferences such as paraphrases and compressions. The entire solution space of possible extractions and QG-generated paraphrases is searched efficiently through use of integer linear programming. The ILP framework allows us to model naturally as constraints, additional requirements such as sentence length, overall summary length, topic coverage and, importantly, grammaticality. We argue that QG is attractive for describing rewrite operations common in summarization. Rather than assuming a strictly synchronous structure over the source and target sentences, QG identifies a “loopy” alignment of parse trees assuming that the target tree is in some way “inspired by” the source tree. A key insight in our approach is to formulate the summarization problem at the phrase level: both QG rules and information extraction operate over individual phrases rather than (as is the norm) sentences. At this smaller unit level, QG rules become more widely applicable and compression falls naturally because only phrases deemed important should appear in the summary. We evaluate the proposed model on headline generation and the related task of image caption generation. However, there is nothing inherent in our formulation that is specific to those two tasks; it is possible for the model to generate longer or shorter summaries, for a single or multiple documents. Experimental results show that our method obtains state-of-the-art performance, both in terms of grammaticality and informativeness for both tasks using the same summarization model.

Journal: EMNLP

[Download PDF](#) [View Graph](#)

### Category

SENTENCE SUMMARY

### Authors

Woodsend, Kristian, Feng, Yanzong, Lapata, Mirella

### Citations

0 Citations

## Citations

### Related Papers

Handling Noisy Queries in Cross Language FAQ Retrieval	Unsupervised Discovery of Negative Categories in Lexicon Bootstrapping	Automatic Discovery of Manner Relations and its Applications
There has been a tremendous growth in the number of new mobile subscribers in the recent past. Most of these new subscribers are from developing countries where mobile is the primary information device. Even for users familiar with computers and the Internet, there is nothing inherent in our formulation that is specific to those two tasks; it is possible for the model to generate longer or shorter summaries, for a single or multiple documents. Experimental results show that our method obtains state-of-the-art performance, both in terms of grammaticality and informativeness for both tasks using the same summarization model.	Automatically acquiring semantic lexicons from text is essential for overcoming the knowledge bottleneck in many NLP tasks, e.g. question answering (Ravichandran and Hovy, 2002). Many of the successful methods follow the unsupervised iterative bootstrap principle.	Extracting semantic relations from text is an important step towards understanding the meaning of text. Many applications that use no semantics, or only shallow semantics, could benefit by having available more text semantics. Recently, there is a growing interest in automatically discovering semantic relations from text, such as manner relations, which are often used in natural language processing tasks.

See More

**Purpose**

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

**Categories**

- MODEL
- TRANSLATION
- LANGUAGE
- AL
- APPROACH
- ET

**Send Your Suggestions**

Suggestion Title

Enter your suggestions here

Copyright © 2023 All Rights Reserved

## Contact Page:

Anyone can contact us by submitting the form and the details will be shared with the admin via email and admin will try to resolve your problem asap. The contact us form user interface is shown below:

Figure 19: User Interface - Contact Page

Contact Us

**TELL US ABOUT YOURSELF\***

Enter First Name  Enter Last Name

**ENTER YOUR E-MAIL\***

e.g. someone@email.com

**ENTER SUBJECT\***

e.g. for telling some problem in website

**ENTER YOUR PHONE NUMBER**

2356970

**MESSAGE\***

Message

**Send Message**

**Purpose**

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

**Categories**

- MODEL
- TRANSLATION
- LANGUAGE
- AL
- APPROACH
- ET

**Send Your Suggestions**

Suggestion Title

Enter your suggestions here

**Submit**

Copyright © 2023 All Rights Reserved

### Author Profile Page:

User can click on any author's name and it will navigate to the author's profile from which viewer can see the whole details of the author. The author profile interface is shown below:

*Figure 20: User Interface - Author Profile*

The screenshot displays the 'Author Profile' section of a research database. At the top, there is a circular profile picture of Hafiz Haseeb Ahmad Butt, followed by his name and contact information: Email: hbhatt.bscs19secs@seecs.edu.pk, Affiliation: National University of Science and Technology, and Research Areas: Neural Science. Below this is a search bar with the placeholder 'Search Paper Title OR Abstract' and a blue 'SEARCH' button.

The first research paper listed is 'Summarizing Contrastive Viewpoints in Opinionated Text'. It includes a brief abstract: 'The amount of opinionated text available online has been growing rapidly, increasing the need for systems that can summarize opinions expressed in such text so that a user can easily digest them. In this paper, we study how to summarize opinionated te'. Below the abstract are two buttons: 'Download PDF' and 'View Graph'.

The second paper is 'Automatically Producing Plot Unit Representations for Narrative Text'. Its abstract states: 'In the 1980s, plot units (Lehnert, 1981) were proposed as a knowledge structure for representing narrative stories and generating summaries. Plot units are fundamentally different from the story representations that preceded them because they focus'. It also features 'Download PDF' and 'View Graph' buttons.

The third paper is 'Turbo Parsers: Dependency Parsing by Approximate Variational Inference'. Its abstract mentions: 'Feature-rich discriminative models that break locality/independence assumptions can boost a parser's performance (McDonald et al., 2006; Huang, 2008; Finkel et al., 2008; Smith and Eisner, 2008; Martins et al., 2009; Koo and Collins, 2010). Often, infer- e'. It includes 'Download PDF' and 'View Graph' buttons.

At the bottom of the interface, there is a footer with a 'Purpose' section containing a short story about a car accident, a 'Categories' section listing MODEL, TRANSLATION, AL, LANGUAGE, DOCUMENT, and TEXT, and a 'Send Your Suggestions' form with fields for 'Suggestion Title' and 'Enter your suggestions here', and a 'Submit' button. Social media sharing icons for Facebook, Twitter, Instagram, and LinkedIn are also present.

## Sign Up:

Researchers have to sign up to use some additional functionalities which they cannot access without signing in as a user. The only logged in users can save articles and can also switch their account to researcher account. The sign-up interface for user is as follows:

*Figure 21: User Interface- Sign Up page*



The sign-up interface features a central illustration of a hand holding a red and blue stylus, writing on a computer monitor. The monitor displays a dark blue screen with a grid pattern. The background is white with abstract blue shapes.

Email address\*

We'll never share your email with anyone else.

Name\*

Password\*

Confirm Password\*

**Sign Up**

### **Sign In:**

Once the user has signed up, s/he has to enter his/her details in the sign in page and after entering the right credentials, user can navigate the other site pages from the user account and access different functionalities. The user sign in interface is as follows:

*Figure 22: User Interface - Sign In*



The sign-in interface features a central illustration of a hand holding a red and blue stylus, writing on a computer monitor. The monitor displays a dark blue screen with a grid pattern. The background is white with abstract blue shapes.

Email address

Forgot password?

**Log In**

Don't have an account? [Register](#)

### **Forgot Password:**

By clicking on forgot password, you will be navigated to the next page from where you can set a new password. The interface is as follows:

*Figure 23: User Interface - Change Password*



The figure shows a user interface for changing a password. On the left, there is a stylized illustration of a hand holding a red stylus, writing on a dark blue rectangular area that appears to be a screen or a document. This is set against a white background with light blue leaf-like shapes scattered around. On the right, there is a form with three input fields and a button. The first field is labeled "Email address\*" and contains the placeholder "Enter Email". The second field is labeled "New Password\*" and contains the placeholder "Enter Password". The third field is labeled "Repeat New Password\*" and contains the placeholder "Confirm Password". Below these fields is a blue rectangular button with the text "Change Password" in white.

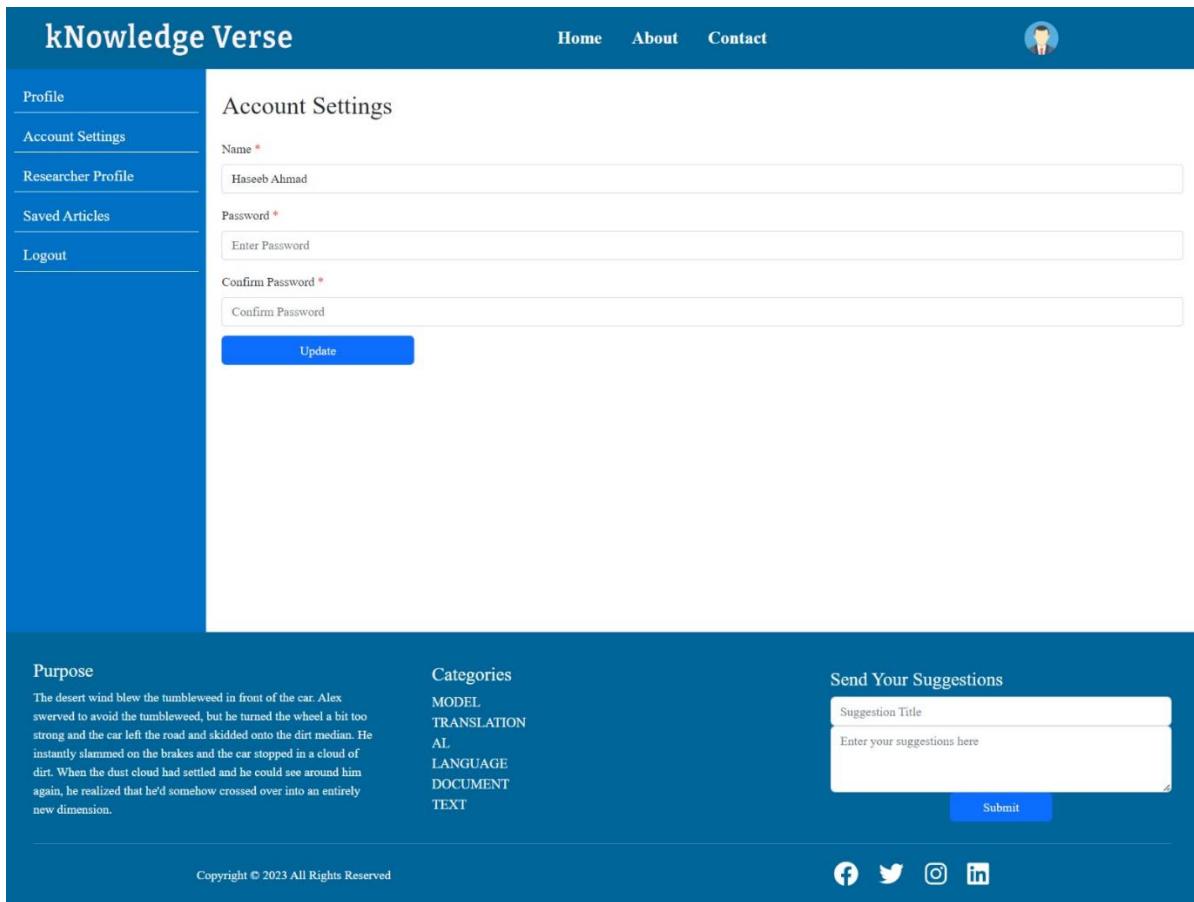
Email address*	<input type="text" value="Enter Email"/>
New Password*	<input type="password" value="Enter Password"/>
Repeat New Password*	<input type="password" value="Confirm Password"/>

**Change Password**

### **Change Name:**

User can change password by going to the settings and user can update name from here.

Figure 24: User Interface - User account settings



### Publisher account Settings:

This is the setting page for user account and user can change his/her password from here and can also promote his account to be a researcher account by filling the mandatory details , user can be a researcher and can upload research papers on this platform and can also papers which are indexed on other sites to this platform. The interface depicting these settings are shown below:

By clicking on researcher profile, user will be navigated to the next page and if user hadn't filled the information below, then empty form will be shown which user has to fill to make user account publisher one. If user account is already publisher, then the already filled information will be shown and user can update it from here.

Figure 25: User Interface - Researcher Information

k**Nowledge Verse**

Home
About
Contact

- [Profile](#)
- [Account Settings](#)
- [Researcher Profile](#)
- [Researcher Information](#)
- [Articles](#)
- [Saved Articles](#)
- [Logout](#)

### Researcher Information

---

Name \*

Add other names

E-Mail for Verification \*

[Add](#)

Areas of Interest

[Add](#)

Affiliation

Affiliation Homepage

Personal Site

[Update](#)

**Purpose**

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

**Categories**

- MODEL
- TRANSLATION
- AI
- LANGUAGE
- DOCUMENT
- TEXT

**Send Your Suggestions**

Suggestion Title

[Submit](#)

Copyright © 2023 All Rights Reserved

80

By clicking on articles, you will move to the new page where the publisher all articles whether they are accepted or rejected will be shown and uploading new articles will be shown.

*Figure 26: User Interface - Publisher Articles*

The screenshot shows the 'kNowledge Verse' user interface. On the left, there is a sidebar with navigation links: Profile, Account Settings, Researcher Profile, Researcher Information, Articles, Saved Articles, and Logout. The main content area is titled 'Research Work Information'. At the top right of this area are 'Home', 'About', 'Contact', and a user icon. Below this is a blue header bar with 'Add Article' and 'Find Article' buttons. A table titled 'Upload Article' lists three articles with columns for Date of Upload, Title, Abstract, and Status. The first two articles are 'ACCEPTED' and the third is 'ACCEPTED'. At the bottom of the table are 'Remove Selected Articles' and a blue 'Submit' button. The 'Purpose' section contains a short story about a car accident. The 'Categories' section lists MODEL, TRANSLATION, AI, LANGUAGE, DOCUMENT, and TEXT. The 'Send Your Suggestions' section has fields for 'Suggestion Title' and 'Enter your suggestions here' with a 'Submit' button. Social media icons for Facebook, Twitter, Instagram, and LinkedIn are at the bottom right.

Date of Upload	Title	Abstract	Status
2023-05-12	Summarizing Contrastive Viewpoints in Opinionated Text	The amount of opinionated text available online has been growing rapidly, increasing the need for systems that can summarize opinions expressed in such text so that a user can easily digest them. In this paper, we study how to summarize opinionat....	ACCEPTED
2023-05-12	Automatically Producing Plot Unit Representations for Narrative Text	In the 1980s, plot units (Lehnert, 1981) were proposed as a knowledge structure for representing narrative stories and generating summaries. Plot units are fundamentally different from the story representations that preceded them because they f....	ACCEPTED
2023-05-12	Turbo Parsers: Dependency Parsing by Approximate Variational Inference	Feature-rich discriminative models that break locality/independence assumptions can boost a parser's performance (McDonald et al., 2006; Huang, 2008; Finkel et al., 2008; Smith and Eisner, 2008; Martins et al., 2009; Koo and Collins, 2010). Often, inf....	ACCEPTED

On selecting a particular article, the article details will be shown and you can remove it from the platform.

Figure 27: User Interface - Published Articles Details

The screenshot displays the kNowledge Verse user interface. On the left, a vertical sidebar menu lists options: Profile, Account Settings, Researcher Profile, Researcher Information, Articles, Saved Articles, and Logout. The main content area shows a selected article titled "Summarizing Contrastive Viewpoints in Opinionated Text". Below the title is a short summary: "The amount of opinionated text available online has been growing rapidly, increasing the need for systems that can summarize opinions expressed in such text so that a user can easily digest them. In this paper, we study how to summarize opinionated te". There are two buttons at the bottom of this section: "Download PDF" and "View Graph". To the right of the summary, publication details are listed: "Published at: EMNLP - 2010", "Citations: 0", and "Topics Covered: OPINIONATED, TEXT, VIEWPOINT". A blue "Remove Articles" button is located below the summary. At the bottom of the page, there are sections for "Purpose" (containing a short story about Alex), "Categories" (listing MODEL, TRANSLATION, AL, LANGUAGE, DOCUMENT, and TEXT), and a "Send Your Suggestions" form. The form includes fields for "Suggestion Title" and "Enter your suggestions here", and a "Submit" button. Social media sharing icons for Facebook, Twitter, Instagram, and LinkedIn are also present.

**kNowledge Verse**

Home   About   Contact

Profile  
Account Settings  
Researcher Profile  
Researcher Information  
Articles  
Saved Articles  
Logout

Articles Selected: 0

Summarizing Contrastive Viewpoints in Opinionated Text

The amount of opinionated text available online has been growing rapidly, increasing the need for systems that can summarize opinions expressed in such text so that a user can easily digest them. In this paper, we study how to summarize opinionated te

Download PDF   View Graph

Published at: EMNLP - 2010  
Citations: 0  
Topics Covered: OPINIONATED, TEXT, VIEWPOINT

Remove Articles

Purpose

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

Categories

MODEL  
TRANSLATION  
AL  
LANGUAGE  
DOCUMENT  
TEXT

Send Your Suggestions

Suggestion Title  
Enter your suggestions here

Submit

Copyright © 2023 All Rights Reserved

f   t   i   l

On clicking the profile button, the publisher profile will be shown in this form:

Figure 28: User Interface - Publisher Profile

k**Nowledge Verse**

- Profile
- Account Settings
- Researcher Profile
- Saved Articles
- Logout



Hafiz Haseeb Ahmad Butt  
other names are: H.H.A.Butt  
**Email:** hbutt.bces19@seecs.edu.pk  
**Affiliation:** National University of Science and Technology  
**Research Areas:** Neural Science

[Home](#)   [About](#)   [Contact](#)



SEARCH

### Summarizing Contrastive Viewpoints in Opinionated Text

The amount of opinionated text available online has been growing rapidly, increasing the need for systems that can summarize opinions expressed in such text so that a user can easily digest them. In this paper, we study how to summarize opinionated te

[Download PDF](#)
[View Graph](#)

### Automatically Producing Plot Unit Representations for Narrative Text

In the 1980s, plot units (Lehnert, 1981) were proposed as a knowledge structure for representing narrative stories and generating summaries. Plot units are fundamentally different from the story representations that preceded them because they focus

[Download PDF](#)
[View Graph](#)

### Turbo Parsers: Dependency Parsing by Approximate Variational Inference

Feature-rich discriminative models that break locality/independence assumptions can boost a parser's performance (McDonald et al., 2006; Huang, 2008; Finkel et al., 2008; Smith and Eisner, 2008; Martins et al., 2009; Koo and Collins, 2010). Often, infer- e

[Download PDF](#)
[View Graph](#)

**Purpose**

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

**Categories**

- MODEL
- TRANSLATION
- AL
- LANGUAGE
- DOCUMENT
- TEXT

**Send Your Suggestions**

Enter your suggestions here

[Submit](#)

[!\[\]\(80c7375ff38e2a4b3bc74ddd4a35f5b7\_img.jpg\)](#)
[!\[\]\(498db244d7218e85aedd3b7099b730dd\_img.jpg\)](#)
[!\[\]\(e80af5b93b66fcb8bca6064c77336fbd\_img.jpg\)](#)
[!\[\]\(de25163876bf2604bfa28a56714bd255\_img.jpg\)](#)

Copyright © 2023 All Rights Reserved

1

83

## **Admin Interfaces:**

Admin has to login with the right credentials and after logging in , admin will be directed to the update researcher page. Every new researcher must be approved by admin before publisher can publish some content. The new publisher requests will be shown in this page. Admin can either accept or reject it.

*Figure 29: User Interface - Publisher Requests*

The screenshot shows the 'kNowledge Verse' Admin interface. On the left, there's a sidebar with navigation links: Account Settings, Author Options (Add Researcher, Update Researcher Status), Update Articles, Statistics, and Logout. The main content area is titled 'Update Researcher Status'. It features a search bar with 'Find' and 'Search' buttons, and a table with one row showing research data. The table columns are: Author E-Mail, Author Name, Affiliation, Profile Link, and Status. The status dropdown is set to 'Active' and has an 'Update' button next to it. Below the table, there's a small number '1' in a blue box. At the bottom of the page, there's a footer with 'Purpose' (a short story about a car accident), 'Categories' (listing MODEL, TRANSLATION, LANGUAGE, AL, APPROACH, ET), 'Send Your Suggestions' (with fields for title and message), and social media sharing icons (Facebook, Twitter, Instagram, LinkedIn). The footer also includes a copyright notice: 'Copyright © 2023 All Rights Reserved'.

Author E-Mail	Author Name	Affiliation	Profile Link	Status
hbutt.bscs19seecs@seecs.edu.pk	Hafiz Haseeb Ahmad Butt	National University of Science and Technology	Open Profile	Active <input type="button" value="Update"/>

Admin will accept or reject the newly uploaded articles. This will be open if user clicks on update articles button. Admin can also see the accepted, deleted articles. The user interface is as follows.

*Figure 30: User Interface - Articles Request*

Year of Publish	Title	Journal	PDF	Status
2010	Summarizing Contrastive Viewpoints in Opinionated Text	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Automatically Producing Plot Unit Representations for Narrative Text	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Handling Noisy Queries in Cross Language FAQ Retrieval	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Learning the Relative Usefulness of Questions in Community QA	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Positional Language Models for Clinical Information Retrieval	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Inducing Word Senses to Improve Web Search Result Clustering	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Improving Translation via Targeted Paraphrasing	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	Soft Syntactic Constraints for Hierarchical Phrase-Based Translation Using Latent Syntactic Distributions	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	A Hybrid Morpheme-Word Representation for Machine Translation of Morphologically Rich Languages	EMNLP	Download PDF	Accept ▾ <button>Update</button>
2010	"Poetic" Statistical Machine Translation: Rhyme and Meter	EMNLP	Download PDF	Accept ▾ <button>Update</button>

1 2 3 4 5

**Purpose**

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

**Categories**

MODEL  
TRANSLATION  
LANGUAGE  
AL  
APPROACH  
ET

**Send Your Suggestions**

Suggestion Title

Enter your suggestions here

Submit

Copyright © 2023 All Rights Reserved

f t i n

If the admin clicks on deleted articles button, then only rejected articles will be shown. The interface would be this one then:

Figure 31: User Interface - Rejected Articles

The screenshot displays the kNowledge Verse user interface. At the top, there is a dark blue header bar with the logo "kNowledge Verse" on the left, and navigation links "Home", "About", and "Contact" on the right. A user profile icon is also present in the top right corner.

The main content area has a white background. On the left, there is a vertical sidebar with a blue background containing several menu items under different sections:

- Account Settings**
- Author Options**
  - Add Researcher
  - Update Researcher Status
- Update Articles**
  - Accepted Article
  - Add Article
  - Deleted Article
- Statistics
- Logout

In the center, there is a search bar with a "Search" button. Below it is a table titled "Rejected Articles". The table has columns: "Year of Publish", "Title", "PDF", "Reason", and "Status". One row is visible, showing data for an article published in 2010 with the title "Turbo Parsers: Dependency Parsing by Approximate Variational Inference". The PDF link leads to "Download PDF", the reason is listed as "some reason", the status is "Rejected", and there is a "Update" button.

Below the table, there is a small blue box containing the number "1".

At the bottom of the page, there is a footer section with the following components:

- Purpose**: A short paragraph about a car accident.
- Categories**: A list of categories: MODEL, TRANSLATION, LANGUAGE, AL, APPROACH, ET.
- Send Your Suggestions**: A form with fields for "Suggestion Title" and "Enter your suggestions here", and a "Submit" button.
- Social media sharing icons for Facebook, Twitter, Instagram, and LinkedIn.
- Copyright notice: "Copyright © 2023 All Rights Reserved".

Admin can also see the statistics of all users and publishers. It can be seen by clicking on the statistics button. The statistics interface is as follows:

Figure 32: User Interface- Admin Statistics

# kNowledge Verse

Home
About
Contact

Profile

[Account Settings](#)
  
[Author Options](#)
  
[Add Researcher](#)
  
[Update Researcher Status](#)
  
[Update Articles](#)
  
[Accepted Article](#)
  
[Add Article](#)
  
[Deleted Article](#)
  
[Statistics](#)
  
[Logout](#)

**Accepted Publications**  
45

**Rejected Publications**  
0

**In-Progress Publications**  
0

**Accepted Authors**  
1

Rejected Authors
Journal Name
Total Publications

Journal Name	Total Publications
EMNLP	45

**Publications Stats**

A bar chart titled "Publications Stats" showing the number of accepted publications in 2010. The y-axis ranges from 0 to 47. A single blue bar reaches the top of the chart at 47, labeled "Accepted".

Year	Accepted Publications
2010	47

**Purpose**

The desert wind blew the tumbleweed in front of the car. Alex swerved to avoid the tumbleweed, but he turned the wheel a bit too strong and the car left the road and skidded onto the dirt median. He instantly slammed on the brakes and the car stopped in a cloud of dirt. When the dust cloud had settled and he could see around him again, he realized that he'd somehow crossed over into an entirely new dimension.

**Categories**

- MODEL
- TRANSLATION
- LANGUAGE
- AL
- APPROACH
- ET

**Send Your Suggestions**

Submit

Copyright © 2023 All Rights Reserved

Facebook
Twitter
Instagram
LinkedIn

87

# **Chapter 7**

## **7. Conclusion and Future Scope**

KnowledgeVerse is a cutting-edge platform that has been developed to provide users with a seamless and user-friendly experience when it comes to searching for scientific papers and articles. The platform is designed to be intuitive, with a simple search bar that allows users to enter a keyword or phrase and receive relevant search results. Additionally, KnowledgeVerse offers a range of filters that can be used to refine search results, including date filters and citation score filters.

One of the key features of KnowledgeVerse is its ability to provide users with detailed information about the papers they are searching for. This includes information such as the paper's title, authors, abstract, and citation score. This information can be invaluable to users who need to quickly assess the relevance and quality of a paper before investing time and resources into reading it.

In addition to its core functionality, KnowledgeVerse also has a number of features that make it a versatile platform for scientific research. For example, users can save search results and papers to their profile, allowing them to easily access them at a later time. Additionally, the platform allows users to collaborate with other researchers by sharing papers and notes with their peers.

Looking to the future, one of the most exciting developments for KnowledgeVerse is the planned development of a crawler. This tool will allow the platform to expand its database by indexing already published papers on different platforms. This will provide users with access to a wider range of scientific literature, further increasing the value and relevance of the platform.

Another potential area for future development is the addition of personalized recommendations. By using machine learning algorithms and user data, KnowledgeVerse could provide users with highly targeted recommendations based on their interests and search history. This could be a game-changer for researchers and students who need to stay up-to-date with the latest research in their field.

Overall, KnowledgeVerse represents an exciting new frontier in the world of scientific research. Its intuitive design, powerful search functionality, and range of features make it an essential tool for anyone who needs to quickly and easily access scientific literature. With the planned addition of a crawler and other new features, the future of KnowledgeVerse is bright and full of potential. We are excited to continue to improve and expand the platform, and we look forward to seeing how it will be used by scientists and researchers around the world.