



**FINAL SEMESTER ASSESSMENT (FSA)**  
**B.TECH. (CSE)**  
**VI SEMESTER**

**UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN  
WITH SOFTWARE ENGINEERING LABORATORY**

**PROJECT REPORT**  
**ON**  
**Knowledge Repository**

SUBMITTED BY

NAME	SRN
------	-----

Rahil N Modi	PES1201802826
Sooryanath I T	PES1201802827
Himanshu Jain	PES1201802828

**JANUARY – MAY 2021**  
**DEPARTMENT OF COMPUTER SCIENCE &**  
**ENGINEERING**  
**RR CAMPUS,**  
**BENGALURU – 560100, KARNATAKA, INDIA**

<b>TABLE OF CONTENTS</b>		
<b>SI.No</b>	<b>TOPIC</b>	<b>PAGE No</b>
	ABSTRACT	3
1.	SOFTWARE REQUIREMENTS SPECIFICATION	4
2.	PROJECT PLAN	17
3.	DESIGN DIAGRAMS	20
4.	MODULE DESCRIPTION	29
5.	TEST CASES	30
6.	SCREENSHOTS OF OUTPUT	37

## **Abstract**

Software development of Knowledge Repository (Wikipedia) has been carried out by following standard principles and design strategies of Object Oriented Analysis and Software Development. A complete end-to-end Software Development Lifecycle (SDLC) has been followed to build the Knowledge Repository. A fully described and extensively detailed software requirement specification (SRS) document and a project plan has been prepared before implementation. Activity diagram, Sequence diagram, Class diagram, Use case diagram and state diagrams have been prepared to thoroughly analyze the features to be implemented and their interactions with the entire system. Web application has been developed for the use cases identified and complete testing of the product is done. Progress of different phases of the lifecycle is tracked using the requirements traceability matrix (RTM).

**SOFTWARE REQUIREMENTS SPECIFICATION**

---

# **Software Requirements Specification**

**for**  
**Knowledge Repository**

**Version 1.0 approved**

**Prepared by Aditya G Burli, Ruthwik HM, Jayant Sipani**

**PES University**

**6th February, 2021**

# Table of Contents

<b>Introduction</b>	<b>1</b>
<b>Purpose</b>	<b>1</b>
<b>Intended Audience</b>	<b>1</b>
<b>Product Scope</b>	<b>1</b>
<b>References</b>	<b>1</b>
<b>Overall Description</b>	<b>1</b>
<b>Product Perspective</b>	<b>1</b>
<b>Product Functions</b>	<b>2</b>
<b>User Classes and Characteristics</b>	<b>2</b>
<b>Operating Environment</b>	<b>2</b>
<b>Design and Implementation Constraints</b>	<b>2</b>
<b>2.6 Assumptions and Dependencies</b>	<b>3</b>
<b>External Interface Requirements</b>	<b>3</b>
<b>User Interfaces</b>	<b>3</b>
<b>Software Interfaces</b>	<b>3</b>
<b>Communications Interfaces</b>	<b>3</b>
<b>Analysis Models</b>	<b>3</b>
<b>4.1 Search Article</b>	<b>3</b>
<b>4.2 Edit Article</b>	<b>4</b>
<b>System Features</b>	<b>5</b>
<b>Edit feature</b>	<b>5</b>
<b>Reading feature( Main )</b>	<b>6</b>
<b>Login feature</b>	<b>6</b>
<b>Other Nonfunctional Requirements</b>	<b>7</b>
<b>Performance Requirements</b>	<b>7</b>
<b>Safety Requirements</b>	<b>7</b>
<b>Security Requirements</b>	<b>7</b>
<b>Software Quality Attributes</b>	<b>8</b>
<b>Business Rules</b>	<b>8</b>
<b>Other Requirements</b>	<b>8</b>

## **Revision History**

Name	Date	Reason For Changes	Version

# **1. Introduction**

## **1.1 Purpose**

*The purpose of this Software Requirement Specification document is to lay out the preliminary requirements and overlay of the Knowledge Repository v1 platform. The scope of this product covered in this SRS document is limited to the basic requirements for the product.*

## **1.2 Intended Audience**

*The intended audience of this SRS document are software developers, academics, digital designers, students, etc. This SRS document goes on to detail the idea and functionality of the product, its background, descriptive analysis, interfaces and other attributes of the design and layout.*

## **1.3 Product Scope**

*The overall scope of the Knowledge Repository software is to provide a basic platform for articles on major subjects to exist with emphasis on academics and general knowledge. Voluntary contribution to articles will be allowed, provided references are made. Anonymity of the editor will be provisioned. Simple user interface to access the contents will be designed.*

## **1.4 References**

- 1) STUB SRS:  
[https://www.cse.iitd.ac.in/~cs5080212/Group08\\_\(STUB\)\\_SRS\\_Version\\_1.1.pdf](https://www.cse.iitd.ac.in/~cs5080212/Group08_(STUB)_SRS_Version_1.1.pdf)
- 2) Software Requirements Specification document with example - Krazytek:  
<https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- 3) Example Software Requirements:  
<https://www.reqview.com/doc/iso-iec-ieee-29148-srs-example>

# **2. Overall Description**

## **2.1 Product Perspective**

*People's source of knowledge has been shifting rapidly from physical medium such as books, newspapers, magazines etc to online platforms and websites. Google has become the best tool for people to gain knowledge. It becomes imperative that a decentralised repository of knowledge is created for people to access information freely and fairly without much central intervention or bias. This goal is what we are trying to achieve with this Knowledge Repository software.*

## **2.2 Product Functions**

*The major functions of the Knowledge Repository software are-*

- *F1- Editing provision,*
- *F2- Provision to report false information,*
- *F3- Database to store the information,*
- *F4- User Login provision,*
- *F5- Basic User Interface,*
- *F6- Downloading articles for offline use*

## **2.3 User Classes and Characteristics**

*Class 1: Readers*

*1.1 Guest Readers- These users do not have an account created, visit the platform infrequently, and have very low active hours. The most likely functionalities used by them are- F5, F6*

*1.2 Active Readers- These users have registered accounts and login to the platform fairly frequently. They browse multiple articles each day on an average and clock a relatively high active hour count. The most likely functionalities used by them are- F2, F4, F5, F6,*

*Class 2: Contributors*

*Here, a policy will be enforced that only registered users with an account created can edit the information present in an article. Contributors are those that have edited one or more articles which have been approved by the moderators( a separate user class that is explained below).*

*Class 3: Moderators*

*Apart from moderation done from our side, a class of users who have made significant approved contributions will be given a choice to become the moderator of a sub topic of their choice. This will further promote the policy of empowering users to have more control over the platform's content.*

## **2.4 Operating Environment**

*The Knowledge Repository software will be operated on a digital web environment. The application source code, backend software, the primary and secondary data will be stored on the cloud. The web environment will be available on all operating systems, all major web browsers within the operating system.*

## **2.5 Design and Implementation Constraints**

*Some of the design and implementation constraints may be relating to the amount of decentralisation of information possible due to credibility issues. Hardware limitations might be the saturation of storage space for the large amount of data that needs to be processed. Another limitation can be the number of topics that can be covered by the software due to lack of moderation. Failure of processing multiple edits of the same article due to packet loss and server overload is plausible. Rectifying expired links or fake references can prove to be a challenge. Summarization of articles that are too long may need to be done to clear memory.*

## **2.6 Assumptions and Dependencies**

*Some of the assumptions for this project are-*

1. Availability of adequate amount of cloud storage to mount all components of the software
2. The initial credibility of information posted on the platform before moderation
3. Authenticity of hyperlinks and references provided in articles.
4. Objectivity and neutrality of user based moderation.

*Dependencies-*

1. Cloud storage
2. Browser/Web compatibility
3. Database application
4. Web hosting services

## **3. External Interface Requirements**

### **3.1 User Interfaces**

*The user interface will use plain HTML, CSS and JavaScript application, without relying on any third party frameworks, and uses the features built into all modern web browsers. It will be a simple and clean user interface.*

### **3.2 Software Interfaces**

*The web-platform can run on any web browser.  
To save the records, we have chosen MySQL DBMS.  
The application will be built on Node.js server.*

### **3.3 Communications Interfaces**

*Communication will take place between the front-end and the back-end, by the use of REST API endpoints, over the HTTP Protocol, supporting all types of web browsers.*

## 4. Analysis Models

### Use Case Diagram

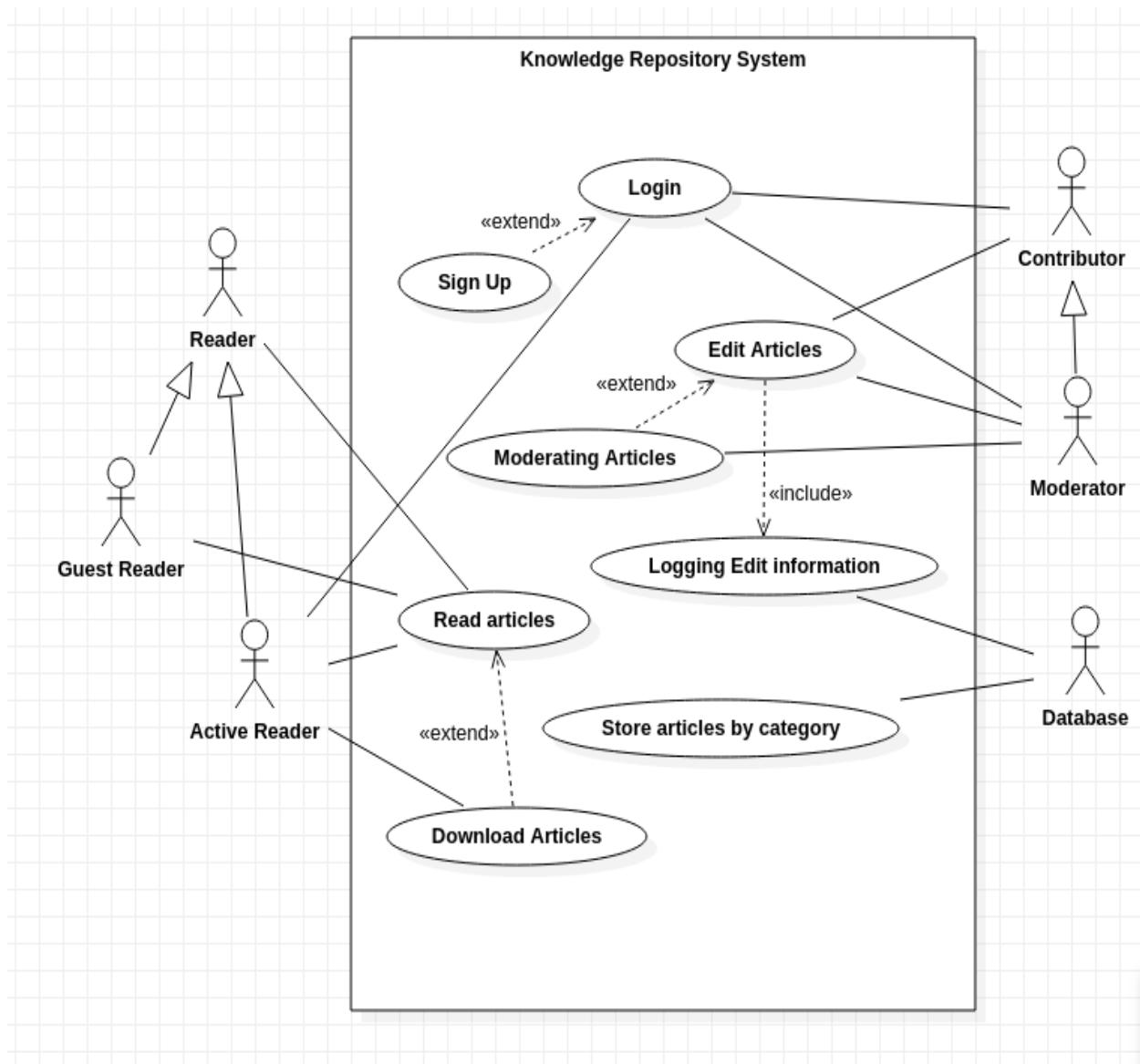


Figure 1.1

#### 4.1 Search Article

<b>Use Case Name</b>	Search Article
<b>Trigger</b>	<i>The Reader assesses the Knowledge Repository</i>
<b>Precondition</b>	<i>The Web is displayed with grids for searching</i>
<b>Basic Path</b>	<p>1. <i>The Reader chooses how to search the Web site. The choices may be by Category or by Keyword.</i></p> <p>2. <i>The system creates and presents a list of all articles by that author in the database.</i></p> <p><i>In step 1, if the Reader selects to search by category, the system creates and presents a list of all categories in the database.</i></p> <p>3. <i>The Reader selects a category.</i></p> <p>4. <i>The system creates and presents a list of all articles in that category in the database.</i></p> <p><i>In step 1, if the Reader selects to search by keyword, the system presents a dialog box to enter the keyword or phrase.</i></p> <p>3. <i>The Reader enters a keyword or phrase.</i></p> <p>4. <i>The system searches the Abstracts for all articles with that keyword or phrase and creates and presents a list of all such articles in the database.</i></p>
<b>Postcondition</b>	<i>The selected article is displayed on the client machine.</i>
<b>Exception Paths</b>	<i>The Reader may abandon the search at any time.</i>
<b>Other</b>	<i>The categories list is generated from the information provided when articles are published and not predefined in the Knowledge Repository.</i>

Table 1.1

## 4.2 Edit Article

<b>Use Case Name</b>	<i>Edit Article</i>
<b>Trigger</b>	<i>The Editor selects to edit a part of the article.</i>
<b>Precondition</b>	<i>The Editor has accessed the Article Manager main screen.</i>
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. <i>The editor selects an article to be edited.</i></li><li>2. <i>He can select any sub-section of the article, and click on the edit option.</i></li><li>3. <i>After successful editing, he saves the changes.</i></li></ol>
<b>Postcondition</b>	<i>The article is properly edited.</i>
<b>Exception Paths</b>	<i>The Editor may abandon the operation at any time.</i>
<b>Other</b>	<i>None</i>

**Table 1.2**

## 4. System Features

### 4.1 Edit feature

#### 5.1.1 Description and Priority

*This feature allows registered users to contribute to the information present on the platform, by editing a portion or multiple parts of an article. This feature is of medium priority as it is considered to be part of our principle of allowing users to be participants in the building of the repository.*

### 5.1.2 Stimulus/Response Sequences

*When the users click on the edit button, they must be directed to the editing interface. In the interface, when the user selects a paragraph to edit, a cursor must show up. After all the edits are done and when the user clicks on the submit button, the edit request must be processed.*

### 5.1.3 Functional Requirements

*R 1.1: Loading of the text present in the sub-part of the article into the editor/text box. This will be achieved with a simple string copy.*

*R 1.2: Processing of the edit- This will be achieved with the use of SQL query in PHP to a database store on the cloud. Errors that occur during the processing will be handled by the PHP script itself.*

## 4.2 Reading feature( Main )

### 5.2.1 Description and Priority

*This is the main feature that caters to the fundamental purpose of creating this product- free and fair access to information and knowledge. Users- both guest and registered will be able to access various articles across a wide range of topics. Users will be able share and download articles as well. This feature is categorised under high priority as it is the main utility/use case of this product.*

### 5.2.2 Stimulus/Response Sequences

*When the users click on the link of any article, they must be redirected to the reading interface. Within the article, if the user clicks on any hyperlink or reference, the user must be redirected to the specified article or referred page. If the user clicks on the report button, then the report wizard must open with options for the user to report the article. If the download button is clicked, then the article in PDF format must be transferred to the local machine.*

### 5.2.3 Functional Requirements

*R 2.1: Reporting an article- A wizard( HTML) with the following options will open- False Information, Outdated Information, NSFW content, Heavy Bias. This report will be stored in a table of the database without the user data and will be used during moderation.*

*R 2.2: Downloading of the article- The text along with its formatting will be written to a separate file which will later be converted to PDF and using an API call transfer it to the local machine.*

*R 2.3 Searching for an article- The text argument given in the search box must be compared with the articles titles present in the database(PHP, MySQL). Top matches must be displayed.*

## 4.3 Login feature

### 5.3.1 Description and Priority

*This feature allows guests users to register themselves and create an account so that they will be able to edit articles. If they are already registered, they can login to their account.*

### 5.3.2 Stimulus/Response Sequences

*When the user clicks on the create account button, they must be directed to the registration page to fill user credentials. When the user clicks on create, the credentials must be processed. When the user clicks on login, it must redirect to the login page and allow the user to login if the credentials match and redirect to their account.*

#### 5.3.3 Functional Requirements

*R 3.1: Account creation- The credentials entered in the form must be inserted to respective tables of the database (PHP, MySQL). Passwords must be hash coded.*

*R 3.2: User login- Credentials must be verified by accessing user data stored in the database(PHP,MySQL).*

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

*This application must be accessible to people with poor internet connectivity as well and hence must be capable of loading the articles on such devices as well. It must have considerable performance and should show no considerable difference when attempted to view on devices with good and poor connectivity. Loading time of articles and redirection to reference hyperlinks must have second level latency.*

### 5.2 Safety Requirements

*Since all these articles are contributed by our user base, some form of basic fact checking must be conducted in order to determine the validity of the information provided by the user on the topic before being updated on our application. Further we must also have some way of the users contributing false information to be held accountable either by disabling their accounts when multiple such offense occur. This will be part of the Terms and Conditions that the user accepts at the time of account creation. Also, this platform is open to all ages and hence uploading of NSFW content or profanity of any kind will be removed and warnings will be issued.*

### 5.3 Security Requirements

*The content hosted on our application will be freely available to the masses. They are free to reuse or share this information. To ensure this the content published must not infringe the proprietary rights of anyone. Further user credentials will have the maximum possible protection on our platform. However we are not accountable for the leaking of user credentials by the user themselves and such users will be held responsible for any misinformation or illegal behaviour on our platform.*

### 5.4 Software Quality Attributes

*The application aims to make available information on various topics to the masses. It provides an easy to use and familiar interface to browse information about various topics. It also aims to*

*have good performance on devices of all kinds. It will also attempt to provide accurate and valid information on various topics. It being a web based application any device with access to the internet and having a browser will be able to make use of the application.*

## 5.5 Business Rules

*Users looking to contribute or modify information on our platform must have an account on our platform. However anyone without an account is free to view and use the information present on our platform. The information contributed by the users must be valid and accurate. Any inaccuracies or misinformation spread by a user is prone to necessary action by us either by warning the user or permanently banning them if multiple such instances have occurred.*

## 6. Other Requirements

### Appendix A: Glossary

### Appendix B: Field Layouts

#### Information required to register the user on the service

Field	Length	Data Type	Description	Is Mandatory
Full Name	60	Alphanumeric with spaces	Name of the user	Y
Username	30	Alphanumeric with underscores	Login credentials of the user	Y
Password	256	Alphanumeric	Password used to login	Y
Age	3	Numeric	Age of the user	Y
Phone number	10	Varchar	Phone number of the user	Y
Email Id	256	Alphanumeric	Email Id of the user	Y

**Table 1.3**

## Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	R1.1	Loading article text for editing purpose	Figure 3.4.1 Figure 3.4.2	Figure 3.4.1 Figure 3.4.2	contributor.html contributor.css contributor.js	UT-17 UT-18 UT-20	UT-22
2	R1.2	Processing of the edit	Figure 3.4.1 Figure 3.4.2	Figure 3.4.1 Figure 3.4.2	contributor.html contributor.css contributor.js	UT-13 UT-14 UT-15	UT-16 UT-25
3	R2.1	Reporting an article	Figure 3.5.1 Figure 3.5.2	Figure 3.5.1 Figure 3.5.2	moderator.html moderator.css moderator.js	UT-26 to UT-35	
4	R2.2	Downloading an article	Figure 3.1 Figure 3.2	Figure 3.1 Figure 3.2	downloadhome.js downloadhome.html	UT-11 UT -12	
5	R2.3	Searching for an article	Figure 3.1 Figure 3.2 Figure 3.6.1 Figure 3.6.2	Figure 3.1 Figure 3.2 Figure 3.6.1 Figure 3.6.2	downloadhome.js Home.js Home.html downloadhome.html	UT-05 UT-06 UT-07 UT-08 UT-09	
6	R3.1	Account creation	Figure 3.2	Figure 3.2	SignIn.js SignIn.html	UT-03 UT-04	
7	R3.2	User login	Figure 3.2	Figure 3.2	SignIn.js SignIn.html	UT-01 UT-02	

## PROJECT PLAN

# PROJECT PLAN- KNOWLEDGE REPOSITORY

### 1: Suitable Lifecycle for the Project

For the Knowledge Repository project, the Software Development Lifecycle(SDLC) has proven to be the most suitable lifecycle that can be chosen. The justification for this owes to the fact that this project by its very nature requires systematic software development. The processes involved in the SDLC is well suited to realise the final outcome of this project and also leads to creation of a quality product. This may not be the case with other lifecycles. Below image depicts the SDLC.

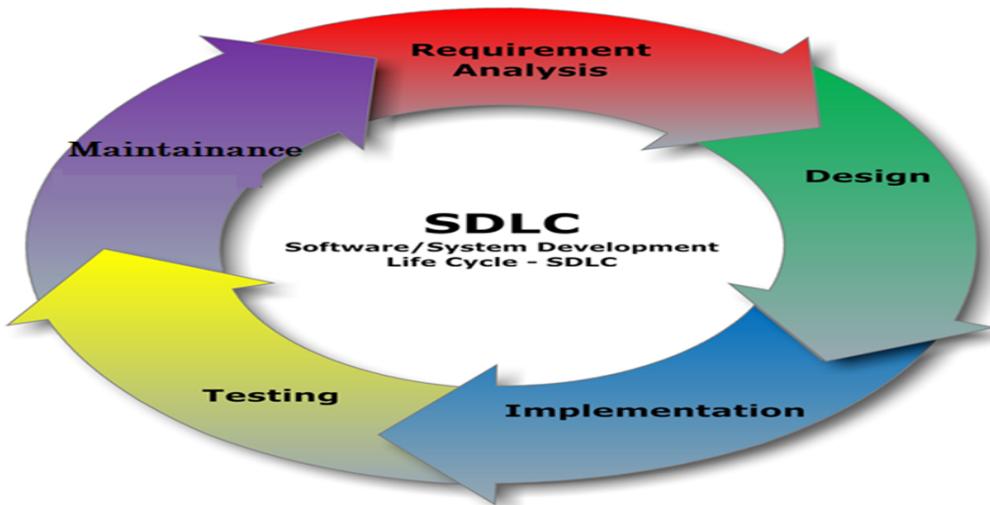


Figure 2.1

### 2: Tools to use at different phases of the SDLC

Requirement Analysis phase - In this phase a planning tool like **KissFlow/Microsoft Project** can be used to determine the different stages of the project, effort required and a timeline for it as well.

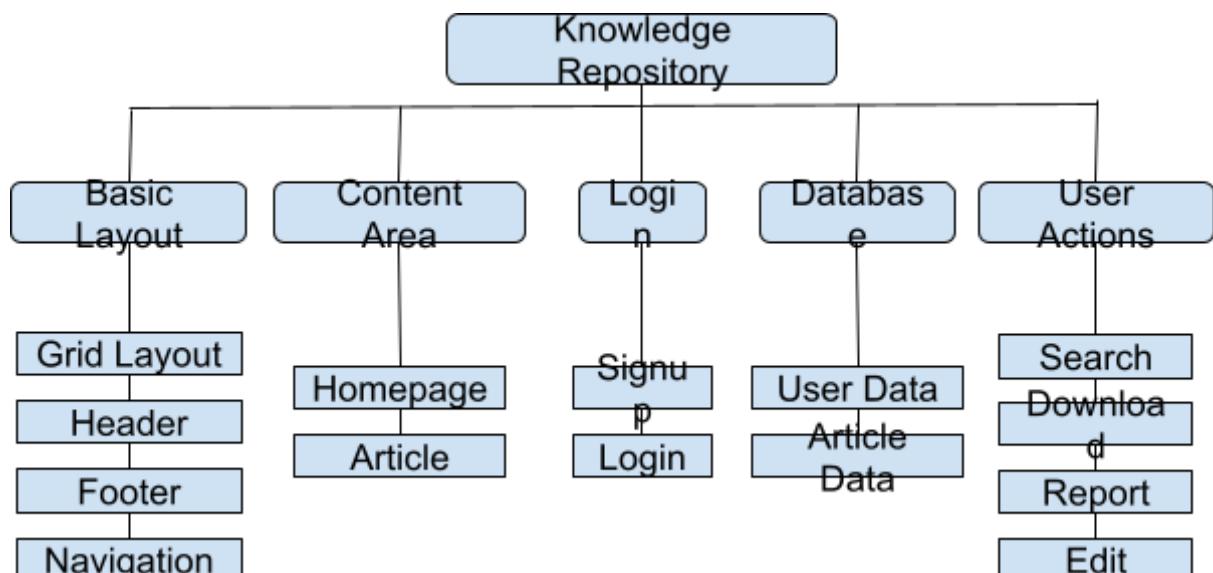
Design phase- In the design phase, an effective design tool like **UIPRINT/StarUML** can be used to chalk out the outlay of the project and decide the basic structure of the components using wireframes, mockups etc.

Implementation phase- During the implementation phase, a development tool like **Visual Studio/Atom IDEs** can be used to build the interfaces, functionalities and also strengthen the software to a large extent.

Testing phase- A testing tool/environment like **web sandboxes** can be used during the testing phase to test all the different components of the software, and eventually fix the bugs that show up.

Maintenance phase- After the final release of the software, when maintenance is being done, version control can be used to smoothly provide patches and updates to improve the product.

### 3: Work Breakdown Structure(WBS)



**Figure 2.2**

### 4: Deliverables

Build components:-

1. Backend Database(User)- The DB that stores user data must be built from scratch.
2. Backend Database(Platform)- Similarly, the DB that stores the data that is on the platform also has to be built.
3. Platform Frontend- The reading UI, editing UI and other webpages of the software need to be built.

4. Platform Backend- Database manipulation to perform account creation, login, accessing articles and recording reports have to be done with a backend scripting.

## 5: Estimation of Effort

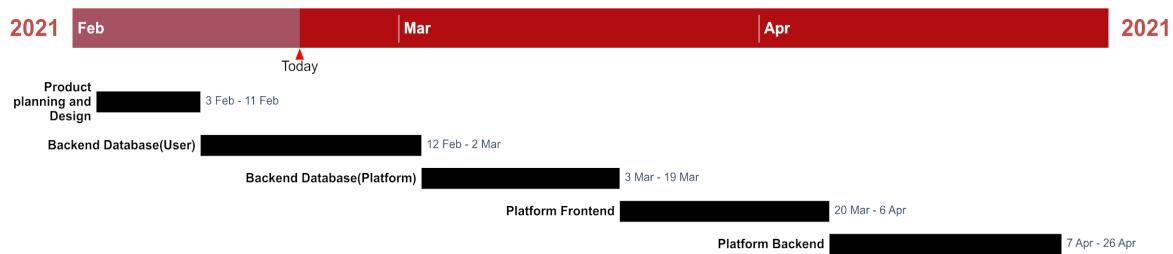
Deliverable 1= 1 person month

Deliverable 2= 1 person month

Deliverable 3= 3 person months

Deliverable 4= 3 person months

## 6: Gantt Chart



**Figure 2.3**

## DESIGN DIAGRAMS

### Class Diagram

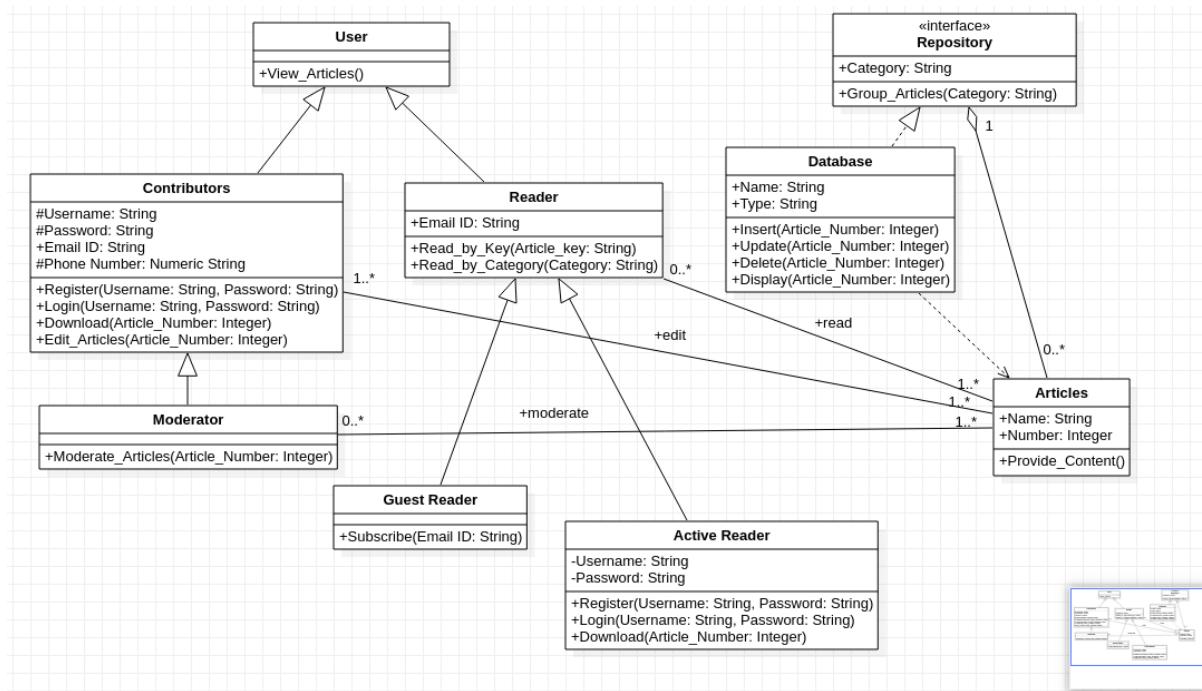
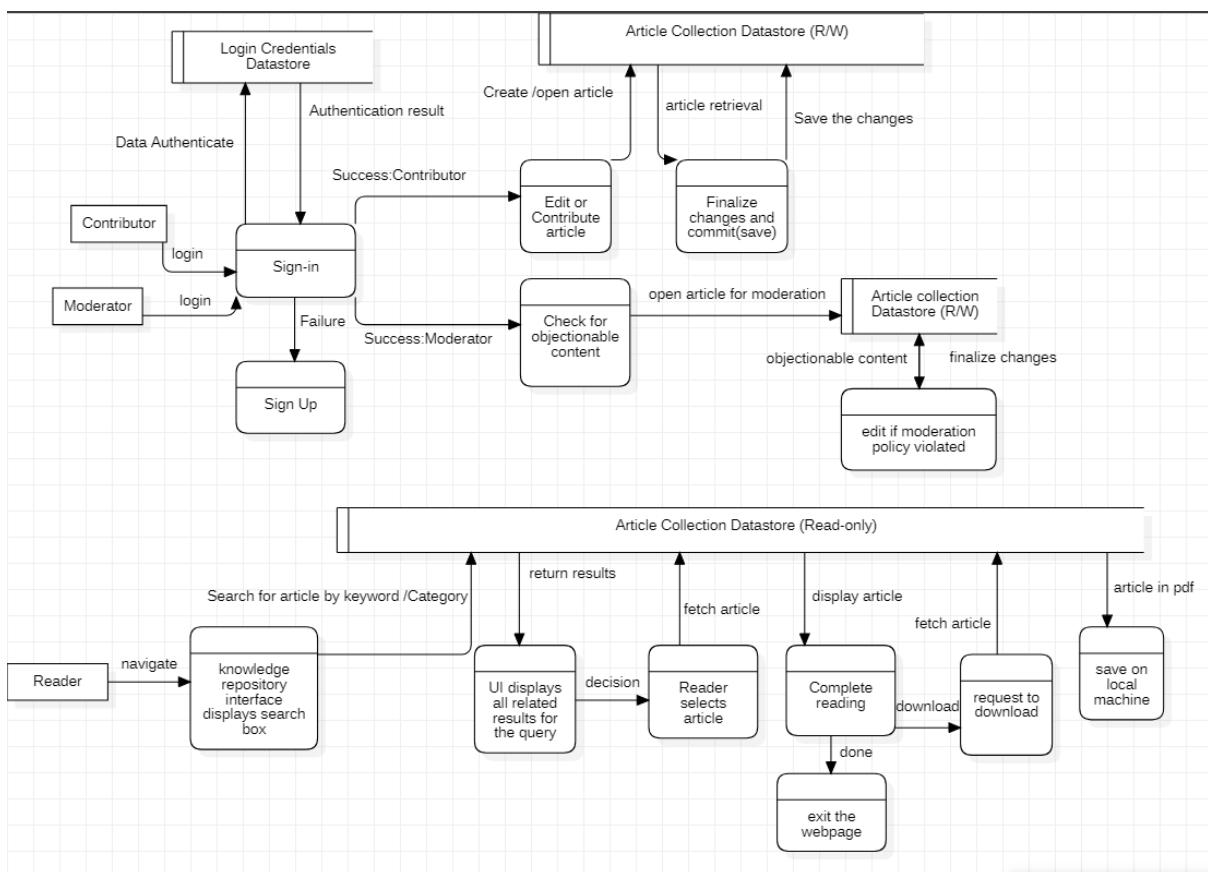


Figure 3.1

The class diagram for the knowledge repository consists of nine classes. The User class is the parent for Contributors and Readers class. Moderator is one of the Contributors. There are two types of Readers: Guest Readers and Active Readers. The Database will store all the Articles posted by the Contributors. Only read access is allowed for the Reader objects. The Repository is an aggregate of all the articles grouped by certain topics.

# System Architecture And Design



**Figure 3.2**

The knowledge repository will have a user interface through which the user can sign up and log in. The contributor is allowed to create and modify the articles. All the articles are validated and approved by the moderator to follow the policies of the system. Once approved, the articles are added to the database through which the reader can retrieve the articles. The interface displays a search box and filters the articles based on the search query, keywords, or topics. The user can request the pdf version of the article and can save it for a later purpose.

## State Diagram

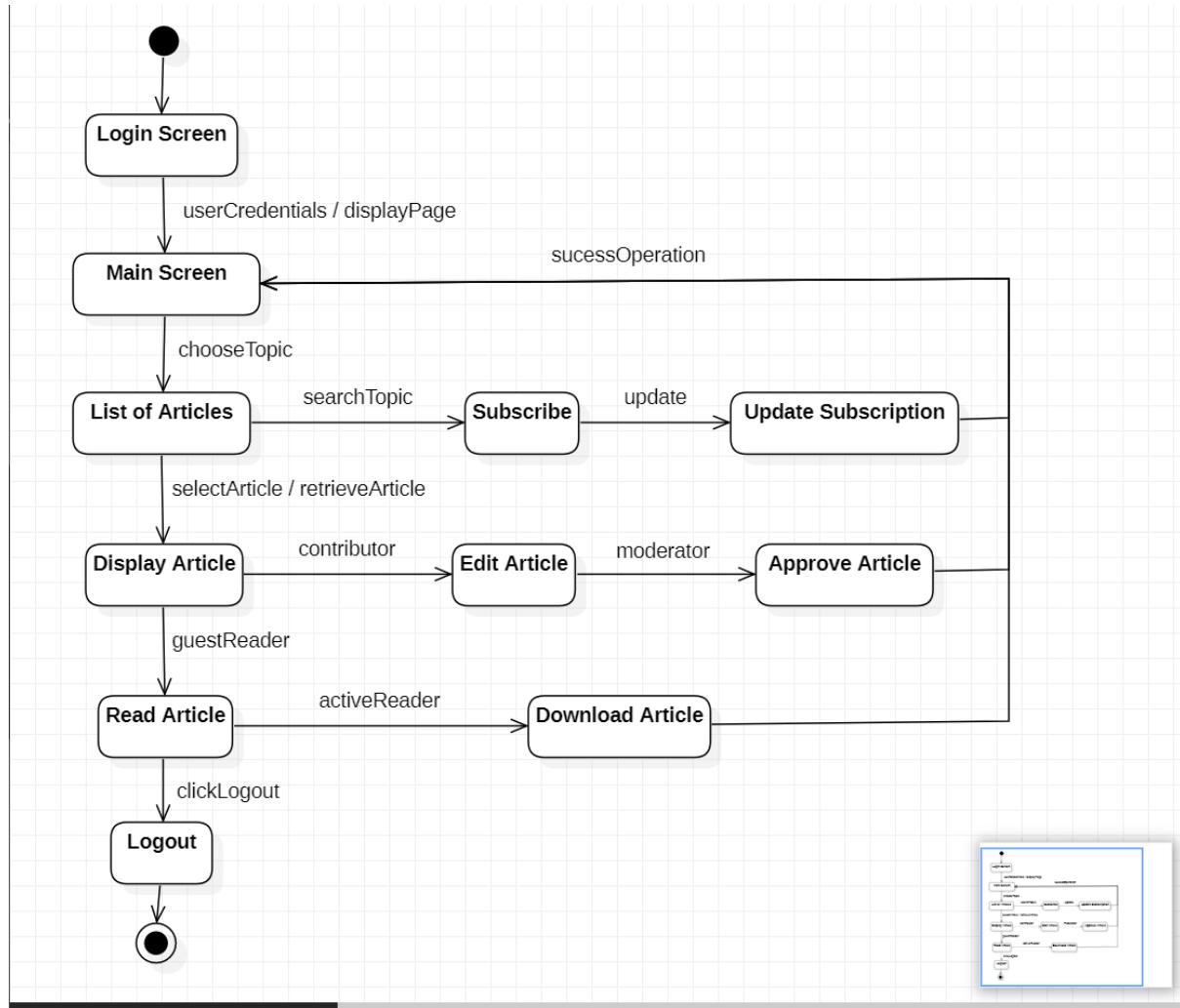


Figure 3.3

## Feature 1: Managing Articles (Edit, Modify and Delete)

Activity Diagram

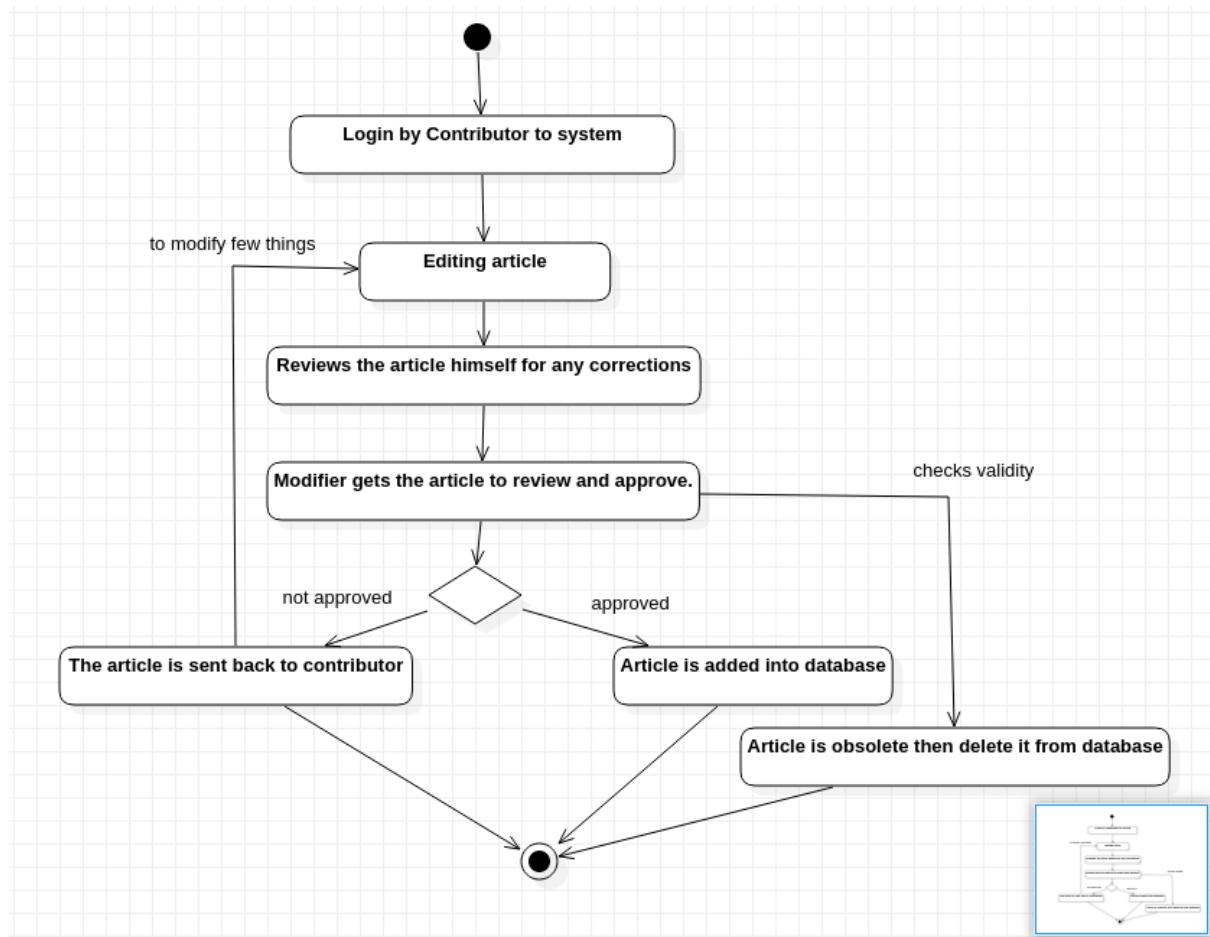
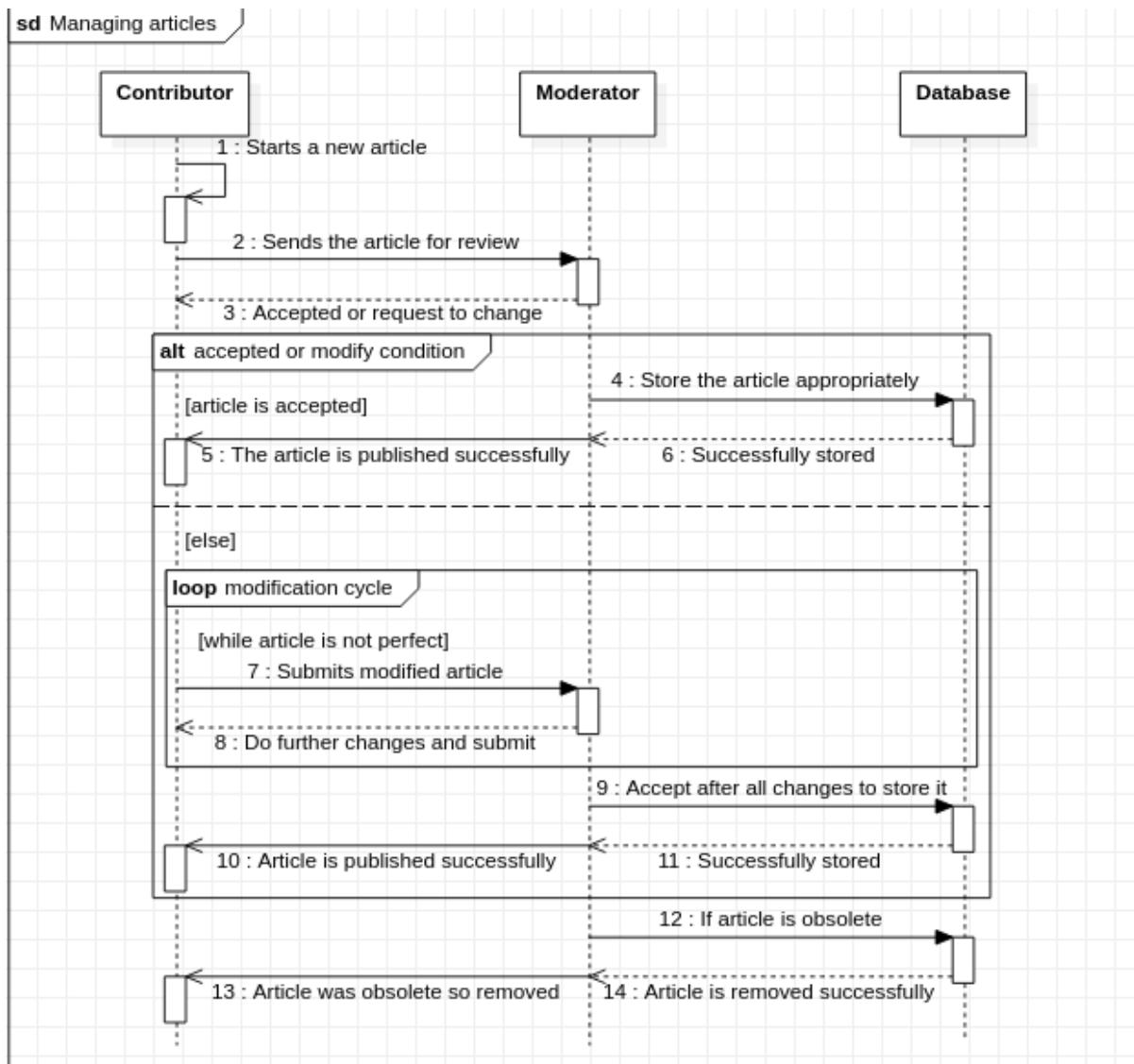


Figure 3.4.1

## Sequence Diagram



**Figure 3.4.2**

## Feature 2: Approval and Review for content changes

Activity Diagram

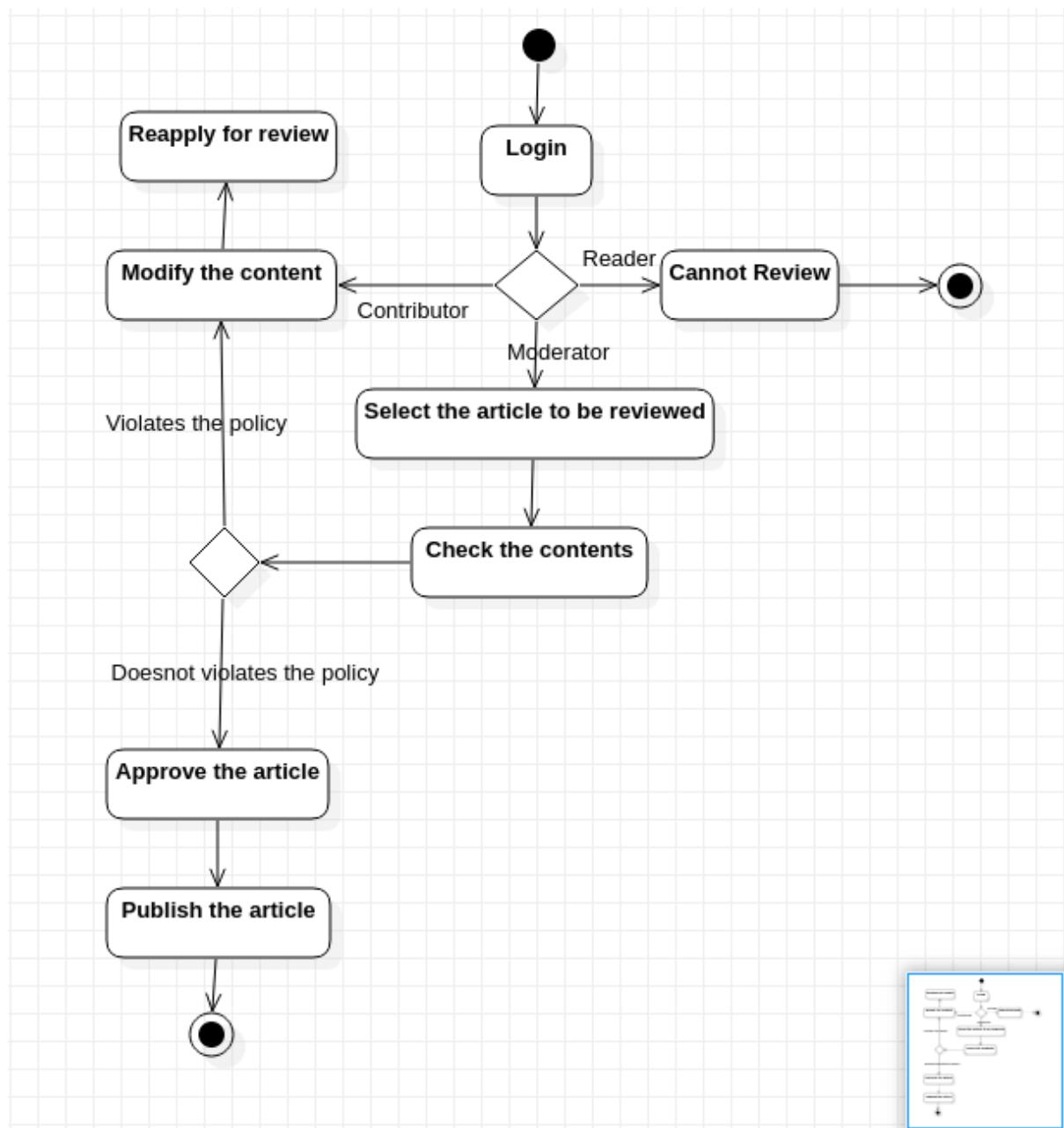


Figure 3.5.1

## Sequence Diagram

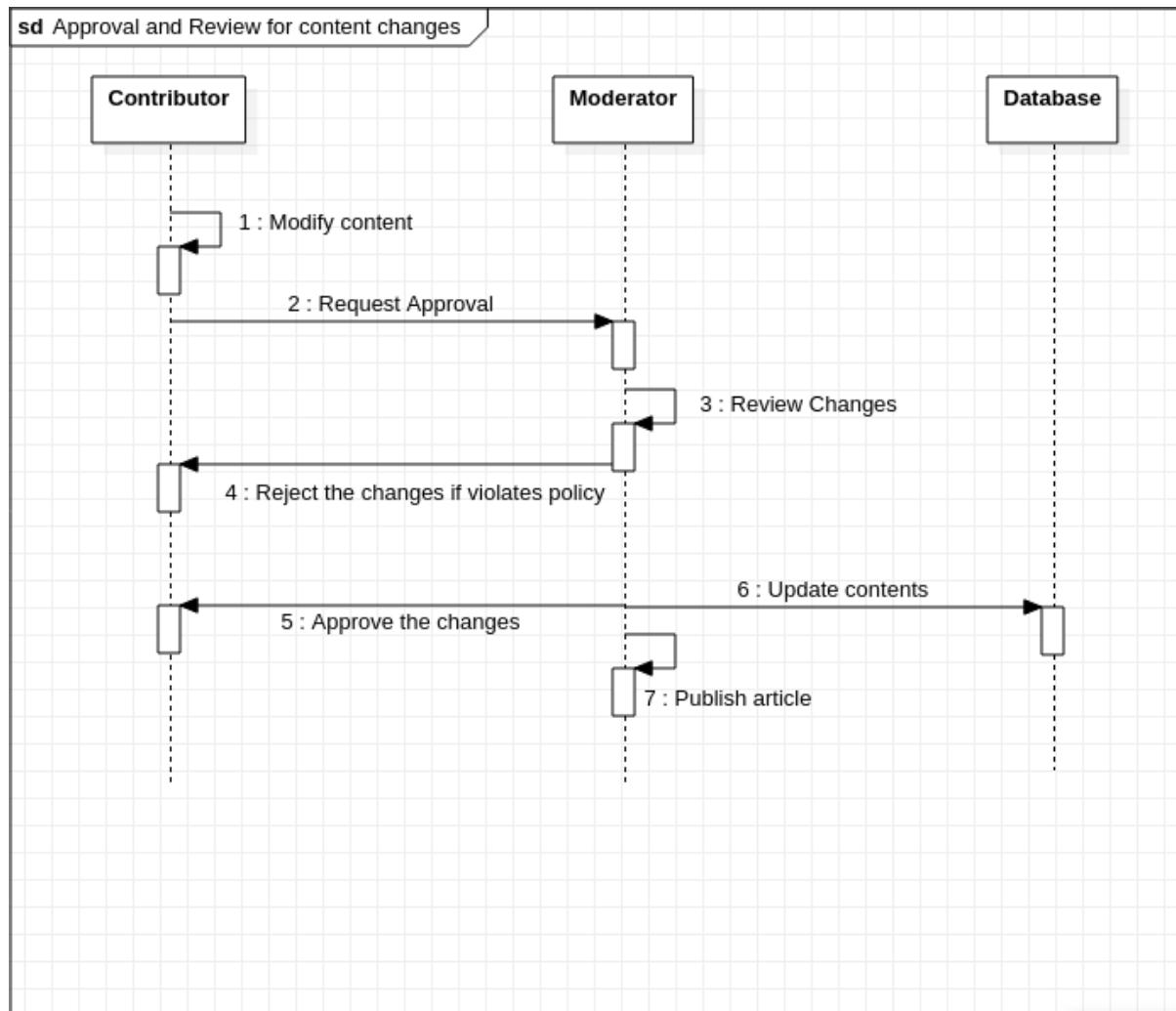
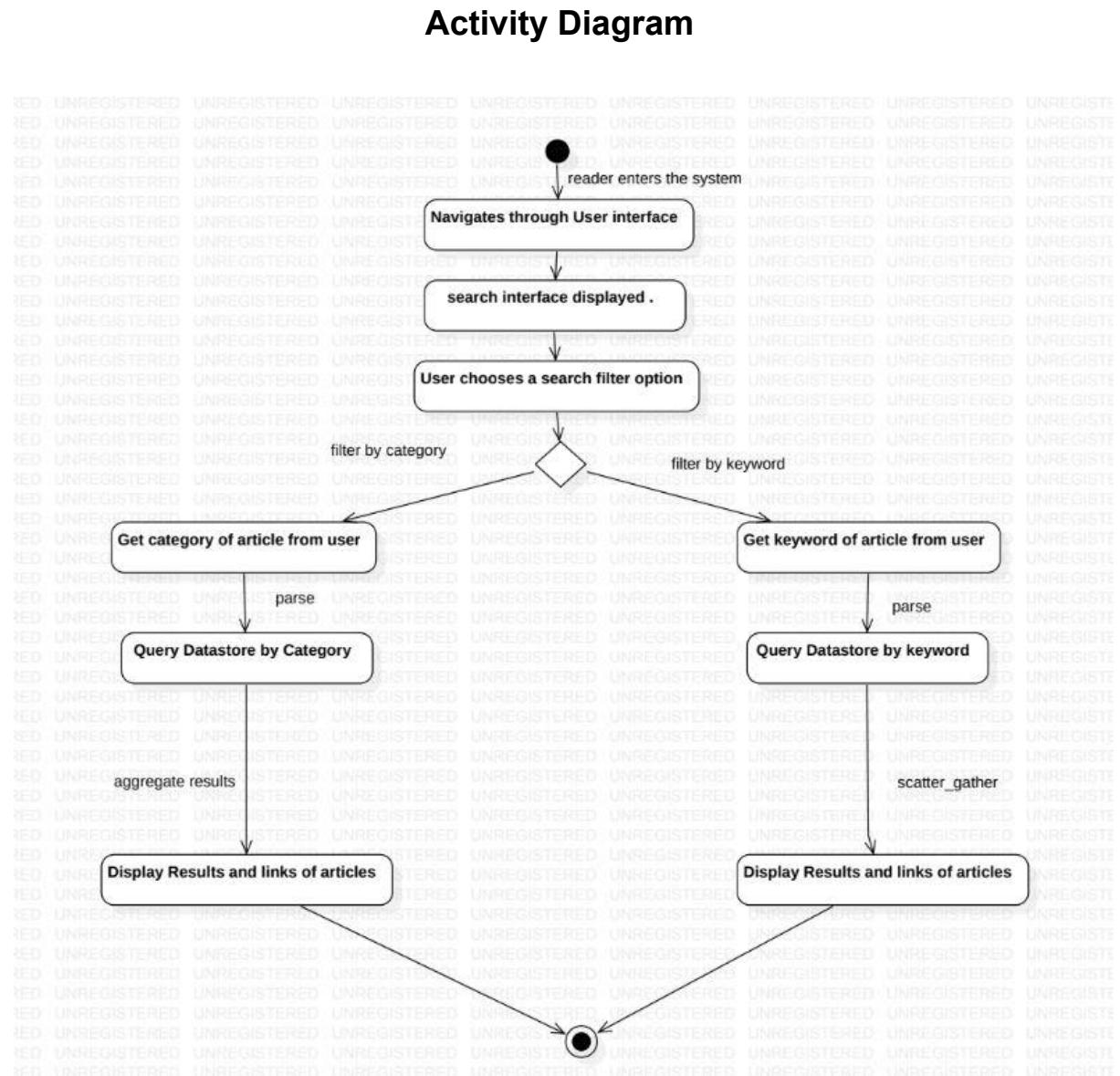


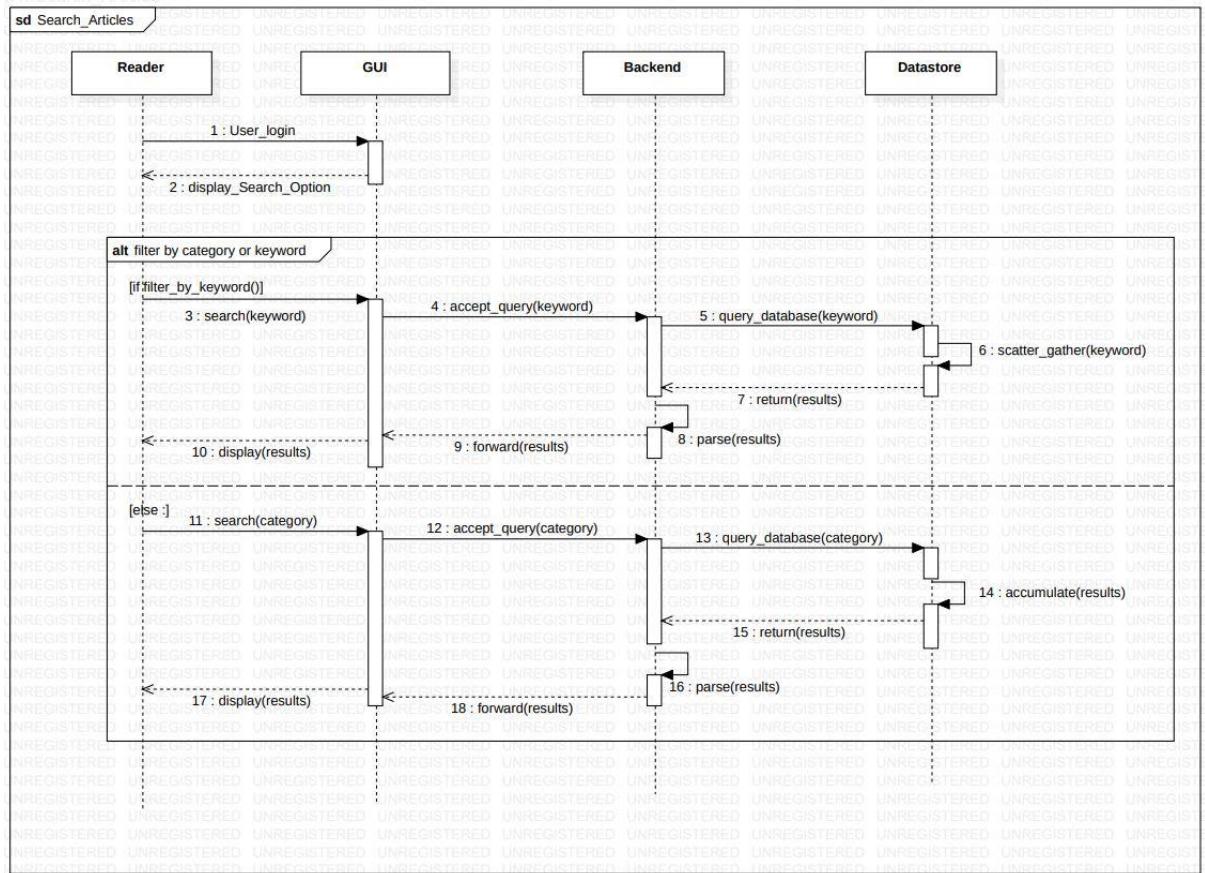
Figure 3.5.2

## Feature 3: Search for articles



**Figure 3.6.1**

## Sequence Diagram



**Figure 3.6.2**

## **MODULE DESCRIPTION**

### **Search and Display articles:**

This is the only module which is exposed to the readers/subscribers. Once the user enters for the search phrase aka search key and chooses the desired option of category based or keyword based search , the information from text area input field is extracted and processed using the business logic for this module at the backend using the javascript methods.The NO-SQL database is matched against the search keyword in the appropriate attributes and the matching results if found are returned , else the appropriate alert message is displayed to the user.

### **Edit and Modify articles:**

The module is used to add new articles in the back store. This is only possible if he logs in the form of a contributor. He has permissions to add and edit articles as required. The articles he had added are to be approved by a moderator before they are viewed by other readers. He has the privilege to view his pending and approved articles. He also gets the required message from the moderator if his article is rejected so that he can modify and submit it again.

### **Approval and Reject module:**

The module is used to approve and reject the documents submitted by the contributors. The moderator will approve the articles if it follows the submission policy. Any document that does not follow the guidelines and standards will be rejected. The moderator has to provide the reason for rejection in the text box. This message will be sent to the contributor and the document will not be added to the approved database. When there are no documents to be approved, the moderator can log out of the system.

# TEST CASES

## Sample Test Cases:

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-01	User Authentication	Role based Authentication and Access to resources	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to the welcome webpage. 2.Choose on <b>Sign In As Privileged User</b> 3.Enter registered username , password and role. 4.Choose Sign In	<b>email :</b> <a href="mailto:sooryanathit@gmail.com">sooryanathit@gmail.com</a> <b>passkey :</b> abcd1234 <b>Role :</b> Subscriber	pop-up displaying “Welcome Sooryana th I T”	pop-up displaying “Welcome Sooryana th I T”	Pass
UT-02	User Authentication	Verifying if the signee is logging in with designated role	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to the welcome webpage. 2.Choose on <b>Sign In As Privileged User</b> 3.Enter registered username , password and role. 4.Choose Sign In	<b>email :</b> <a href="mailto:sooryanathit@gmail.com">sooryanathit@gmail.com</a> <b>passkey :</b> abcd1234 <b>Role :</b> Moderator	pop-up displaying “INVALID ROLE”	pop-up displaying “INVALID ROLE”	Pass
UT-03	User Authentication	Verifying Credentials of the signee	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to welcome webpage. 2.Choose on <b>Sign In As Privileged User</b> 3.Enter registered username , password and role. 4.Choose Sign In	<b>email :</b> <a href="mailto:sooryanathit@gmail.com">sooryanathit@gmail.com</a> <b>passkey :</b> wrongpass <b>Role :</b> Moderator	pop-up displaying “The password is invalid or the user does not have a password. Sign Up As A New User!!!”	pop-up displaying “The password is invalid or the user does not have a password .Sign Up As A New User!!!”	Pass
UT-04	User Authentication	Verifying Data Sufficiency to Sign-in /Sign -up	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to welcome webpage. 2.Choose on <b>Sign In As Privileged User</b> 3.Enter registered username , password and role. 4.Choose Sign In	<b>email : “ ”</b> <b>passkey :</b> wrongpass <b>Role : “ ”</b>	pop-up displaying “Fill All The Fields”	pop-up displaying “Fill All The Fields”	Pass
UT-05	Search Articles	Verifying Data Sufficiency to Search and Display Articles	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to	1.Navigate to the welcome webpage. 2.Either Choose <b>Continue As A Reader</b> 3.Choose on <b>Sign In As a Privileged User .</b>	search phrase : “ ”	pop-up displaying <b>Fill in The search Field</b>	pop-up displaying <b>Fill in The search Field</b>	Pass

			Web/Internet	4.Enter registered username , password and role 4.Choose Sign In				
UT-06	Search Articles (Categorical)	Search by Category and display results to reader on UI based on privilege levels	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to welcome webpage. 2.Either Choose <b>Continue As A Reader</b> 3.Choose on <b>Sign In As a Privileged User</b> . 4.Enter registered username , password and role 4.Choose Sign In 5.Enter Name of the article 6.Choose <b>Category Filter Feature</b>	search phrase = “OOAD”	1.display list of articles with category tag <b>OOAD</b> . Namely "SRS Document For Knowledge Repo",  2.Display in either read only format or download able link according to privilege level	display of list of articles with category OOAD , namely <b>SRS Document For Knowledge Repo</b>	Pass
UT-07	Search Articles (Categorical)	Displaying alert regarding unavailable articles , if search is unsuccessful	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to the welcome webpage. 2.Either Choose <b>Continue As A Reader</b> 3.Choose on <b>Sign In As a Privileged User</b> . 4.Enter registered username , password and role 4.Choose Sign In 5.Enter Name of the article 6.Choose <b>Category Filter Feature</b>	search phrase = “Big Data”	1.display pop up indicating <b>No Matching Results Found</b>  2. Auto clear the search field	1.display pop up indicating <b>No Matching Results Found</b>  2. Auto clear the search field	Pass
UT-08	Search Articles (Keyword)	Search by Keyword and display results to reader on UI based on privilege levels	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to the welcome webpage. 2.Either Choose <b>Continue As A Reader</b> 3.Choose on <b>Sign In As a Privileged User</b> . 4.Enter registered username , password and role 4.Choose Sign In 5.Enter Name of the article 6.Choose <b>Category Keyword Filter</b>	search phrase = “Network Security”	1.display list of articles with the corresponding keyword Namely " <b>SEED LABS Manual for CNS</b> "  2.Display in either read only format or	display of list of articles with keyword “Network Security”  Namely " <b>SEED LABS Manual for CNS</b> "	Pass

						download able link as per privilege level		
UT-09	Search Articles (Keyword)	Displaying alert regarding unavailable articles , if search is unsuccessful	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to welcome webpage. 2.Either Choose <b>Continue As A Reader</b> 3.Choose on <b>Sign In As a Privileged User</b> . 4.Enter registered username , password and role 4.Choose Sign In 5.Enter Name of the article 6.Choose <b>Category</b> Keyword Filter	search phrase = “Operating Systems”	1.display pop up indicating  <b>No Matching Results Found</b>  2. Auto clear the search field	1.display pop up indicating  <b>No Matching Results Found</b>  2. Auto clear the search field	Pass
UT-10	Search Articles (UI reset)	Clear previous results before initiating new search	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to the welcome webpage. 2.Either Choose <b>Continue As A Reader</b> 3.Choose on <b>Sign In As a Privileged User</b> . 4.Enter registered username , password and role 4.Choose Sign In 5.Enter Name of the article 6.Choose <b>Category</b> Keyword Filter	search phrase = “OOAD”	1.clear pre existing search results and display current results only	1.The search and display history is cleared and only the current search results are displayed	Pass
UT-11	Search Articles (Download features)	Disable download for unsigned users	1.Access to Web browser (Chrome , MS Edge etc) 2.Access to Web/Internet	1.Navigate to the welcome webpage. 2.Choose <b>Continue As A Reader</b> 3Enter name of the article 4. Choose category filter	search phrase = “OOAD”	1.Display the article pdf if it exists , in a read only format and disable download options	The article is displayed as an iframe and download pop up is blocked	Pass
UT-12	Search Articles (Download features)	Enable download for signed users	Sign in as privilege d user	3.Enter name of the article 4. Choose category filter	search phrase = “Network security”	1.Display the link to article and enable pop up options to download the article from source address	The article’s link is displayed and on clicking it the user is navigated to download options	Pass

							for the article	
UT-12	Search Articles (Keyword)	Search Articles produces no output when provided with invalid input but the right procedures are carried out	1.Access to Web browser 2.Access to Internet 3. <b>Sign In as a subscriber</b>	1.Sign in as subscriber 2.Fill in the search field	search_phrase : Dummy Category option	1.display list of articles with the corresponding keyword Namely	1.display saying "NO matching articles found"	Fail
UT-13	Upload Documents	As a contributor have <b>access to upload documents</b>	1.Access to Web browser 2.Access to Internet 3. <b>Sign In as a contributor</b>	1: Sign In as a Contributor 2: <b>Enter all the fields.</b> 3: Click Upload	<b>Name:</b> SRS <b>Category:</b> OOAD <b>Keyword:</b> SRS <b>Link:</b> <a href="#">hyperlink</a>	Document successfully uploaded in the database and alert message "Document Uploaded Successfully" is shown.	Document successfully uploaded in the database and alert message "Document Uploaded Successfully" is shown.	Pass
UT-14	Upload Documents	If <b>category is not provided</b> while uploading	1.Access to Web browser 2.Access to Internet 3.Sign In as a contributor	1: Sign In as a Contributor 2: <b>Enter all the fields except the category.</b> 3: Click Upload	<b>Name:</b> SRS <b>Category:</b> - <b>Keyword:</b> SRS <b>Link:</b> <a href="#">hyperlink</a>	Alert saying fill all the fields and document not uploaded.	Alert saying fill all the fields and document not uploaded.	Pass
UT-15	Upload Documents	If <b>name is not provided</b> while uploading	1.Access to Web browser 2.Access to Internet 3.Sign In as a contributor	1: Sign In as a Contributor 2: <b>Enter all the fields except the name.</b> 3: Click Upload	<b>Name:</b> - <b>Category:</b> OOAD <b>Keyword:</b> SRS <b>Link:</b> <a href="#">hyperlink</a>	Alert saying fill all the fields and document not uploaded.	Alert saying fill all the fields and document not uploaded.	Pass
UT-16	Upload Documents	If <b>link or keyword is not provided</b> while uploading	1.Access to Web browser 2.Access to Internet 3.Sign In as a contributor	1: Sign In as a Contributor 2: <b>Enter all the fields except the link or keyword.</b> 3: Click Upload	<b>Name:</b> SRS <b>Category:</b> OOAD <b>Keyword:</b> SRS <b>Link:</b> - (or) <b>Name:</b> SRS <b>Category:</b> OOAD <b>Keyword:</b> - <b>Link:</b> <a href="#">hyperlink</a>	Alert saying fill all the fields and document not uploaded.	Alert saying fill all the fields and document not uploaded.	Pass

UT-17	Edit (Modify ) Documents	To <b>edit the document</b> , and reupload it.	1.Access to Web browser 2.Access to Internet <b>3.Document should be rejected from moderator</b>	1: Rejection message is received from moderator, 2: Reupload document after changes 3: Click Upload	<b>Name:</b> SRS <b>Category:</b> OOAD <b>Keyword:</b> SRS <b>Link:</b> <a href="#"><u>hyperlink uploaded</u></a>	Upload the article in the <b>pending queue</b> and an <b>alert message is seen.</b>	Upload the article in the <b>pending queue</b> and an <b>alert message is seen.</b>	Pass
UT-18	Search pending articles	To <b>search all articles that belong to that contributor in the pending list.</b>	1.Access to Web browser 2.Access to Internet <b>3. Sign In as a contributor</b>	1. Sign In as a Contributor 2: <b>Enter category of requested documents</b> 3: Click Pending articles button	<b>Category:</b> OOAD	Get all articles with <b>category OOAD belonging to that contributor from the pending list</b>	Get all articles with <b>category OOAD belonging to that contributor from the pending list.</b>	Pass
UT-19	Search approved articles	To <b>search all articles that belong to that contributor in the approved list.</b>	1.Access to Web browser 2.Access to Internet <b>3. Sign In as a contributor</b>	1. Sign In as a Contributor 2: <b>Enter category of requested documents</b> 3: Click Approved articles button	<b>Category:</b> OOAD	Get all articles with <b>category OOAD belonging to that contributor from the pending list.</b>	Get all articles with <b>category OOAD belonging to that contributor from the pending list.</b>	Pass
UT-20	Search pending articles	If <b>no pending articles are present in pending queue</b>	1.Access to Web browser 2.Access to Internet <b>3. Sign In as a contributor</b> <b>4. No articles in pending queue</b>	1: Sign In as a Contributor 2: <b>Enter category of requested documents</b> 3: Click Pending articles button	<b>Category:</b> CNS	Display alert message that “ <b>No articles in pending list</b> ”	Display alert message that “ <b>No articles in pending list</b> ”	Pass

UT-21	Search approved articles	If no approved articles are present in pending queue	1.Access to Web browser 2.Access to Internet 3. Sign In as a contributor 4. No articles in database	1: Sign In as a Contributor 2: Enter category of requested documents 3: Click Approved articles button	Category: CNS	Display alert message that “No articles in approved list.”	Display alert message that “No articles in approved list.”	Pass
UT-22	Search pending or approved articles	If no pending or approved articles of given category is present	1.Access to Web browser 2.Access to Internet 3. Sign In as a contributor 4. No articles of a given category are present.	1: Sign In as a Contributor 2: Enter category of requested documents 3: Click the Pending or Approved articles button as per the requirement.	Category: CNS	Display alert message that “No articles are present in the specified category.”	Display alert message that “No articles are present in the specified category.”	Pass
UT-23	Search articles and Upload articles	Earlier data entered in the form and its displayed results are to be cleared	1.Access to Web browser 2.Access to Internet 3. Sign In as a contributor	1: Sign In as a Contributor 2: If we want to upload or search for new articles then we press the clear button on screen.	Press clear button	Clears all the fields and results shown while searching articles previously	Clears all the fields and results shown while searching articles previously	Pass
UT-24	Search articles	When articles are searches if pending or approved button is clicked multiple times then it clears original results and reproduces new results	1.Access to Web browser 2.Access to Internet 3. Sign In as a contributor 4. Already must have searched something	1: Sign In as a Contributor 2: Enter category of requested documents 3: Click the Pending or Approved articles button as per the requirement.	Category: CNS	Clears all previous displayed articles and reproduces new results as per the query.	Clears all previous displayed articles and reproduces new results as per the query.	Pass
UT-25	Download Articles	Contributors can download his articles in case he doesn't	1.Access to Web browser 2.Access to Internet	1: Sign In as a Contributor 2: Enter category of requested documents	Category: OOAD Open SRS of OOAD which is to be downloaded	The contributor is able to download the article	The contributor is able to download the article	Pass

		<b>have a copy</b>	<b>3. Sign In as a contributor</b>	<b>3: Click the Pending or Approved articles button as per the requirement. 4. Open the required article and download it</b>	<b>and press the download button.</b>	<b>to his local machine</b>	<b>article to his local machine</b>	
UT-26	Approve module	To test the entry created in the “read_articles” table	User must be a moderator	1: Open the document submitted 2: Review the document 3: Click Approve	Contributor submitted document	Entry created in read_articles table and pop up message displayed	Entry created in read_articles table and pop up message displayed	Pass
UT-27	Approve module	To test the entry deleted from the “pending_articles” table	User must be a moderator	1: Open the document submitted 2: Review the document 3: Click Approve	Contributor submitted document	Entry deleted from pending_articles table and pop up message displayed	Entry deleted from pending_articles table and pop up message displayed	Pass
UT-28	Reject module	Moderator is able to provide reasons for rejection	User must be a moderator	1: Review the document 2: Enter the comments for rejection 3: Text box should accept the comments	Contributor submitted document	Contributor is able to provide comments for rejection	Contributor is able to provide comments for rejection	Pass
UT-29	Reject module	To test the entry deleted from the “read_articles” table	User must be a moderator	1: Open the document submitted 2: Review the document 3: Provide feedback 3: Click Reject	Contributor submitted document	Entry deleted from read_articles table	Entry deleted from read_articles table	Pass
UT-30	Reject module	To test the entry deleted from the “pending_articles” table	User must be a moderator	1: Open the document submitted 2: Review the document 3: Provide feedback 3: Click Reject	Contributor submitted document	Entry deleted from pending_articles table	Entry deleted from pending_articles table	Pass
UT-31	Reject module	Test if the user is provided with the rejection message	User must have submitted the document that violates the guidelines	1: Moderator provides the rejection message 2: The message appears to the user as a pop up	Invalid document submitted	Pop up message displayed to the user stating the reasons for rejection	Pop up message displayed to the user stating the reasons for rejection	Pass
UT-32	Moderator module	Test to display all the pending articles to be	User must be a moderator	1: Login to the application using moderator credentials 2: Click Sign In	All the pending articles submitted by moderator.	moderator.html page displays all the pending	moderator.html page displays all the	Pass

		reviewed by the moderator		3: Page loads all the pending articles	the contributors	articles retrieved from the firebase	pending articles retrieved from the firebase	
UT-33	Moderator module	Test if the updated list of pending articles is displayed	User must be a moderator	1: Moderator must approve/reject the document 2: The page automatically reloads 3: Updated list of pending articles is displayed	All the pending articles submitted by the contributors	Web page displays all the pending articles after approval/rejection is performed	Web page displays all the pending articles after approval/rejection is performed	Pass
UT-34	Moderator module	Verify if all the pending articles are loaded	User must be a moderator	1: Moderator must login 2: The data from firebase must be loaded 3: Tester must verify if the entries in the firebase and the elements displayed are the same	All the pending articles submitted by the contributors	Correct set of pending articles are displayed on the web page	Correct set of pending articles are displayed on the web page	Pass
UT-35	Moderator module	Test if the entries created in each element reflects the correct details	Login as moderator	1: User credentials must match the moderator details 2: Details of Title, Contributor name, and Link of the document must be retrieved from the pending_articles table 3: Correctly aligned on the web page	Pending articles submitted by one of the contributors	Details of the Title, Contributor name, and Link of the document are displayed	Details of the Title, Contributor name, and Link of the document are displayed	Pass

**Table 5.1**

## SCREENSHOTS OF OUTPUT

### The Firebase Realtime NO-SQL database

The screenshot shows the Firebase Realtime Database console. The left sidebar includes sections for Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning, Release & Monitor, Analytics, Engage, and Extensions. The main area is titled "Realtime Database" and shows a tree view of the database structure:

```
knowledge-repository-3e9e8-default-rtdb
  +-- id: 2
  +-- read_articles
  +-- user_count: 3
  +-- users
```

Below the tree view, it says "Database location: United States (us-central1)".

Figure 6.1

This screenshot shows the same Firebase Realtime Database interface as Figure 6.1, but with more detailed data under the "read\_articles" node:

```
knowledge-repository-3e9e8-default-rtdb
  +-- id: 2
  +-- read_articles
    +-- 0
      +-- Category: "OOAD"
      +-- Contributor: "rahilmodi123@gmail.com"
      +-- Keyword: "SRS"
      +-- Name: "SRS Document For Knowledge Repo"
      +-- location: "https://drive.google.com/file/d/13C5ERbwQdicELu... "
    +-- 1
    +-- 2
    +-- 3
  +-- user_count: 3
  +-- users
```

Below the tree view, it says "Database location: United States (us-central1)".

Figure 6.2

Use case : SEARCH ARTICLES

## Welcome Page

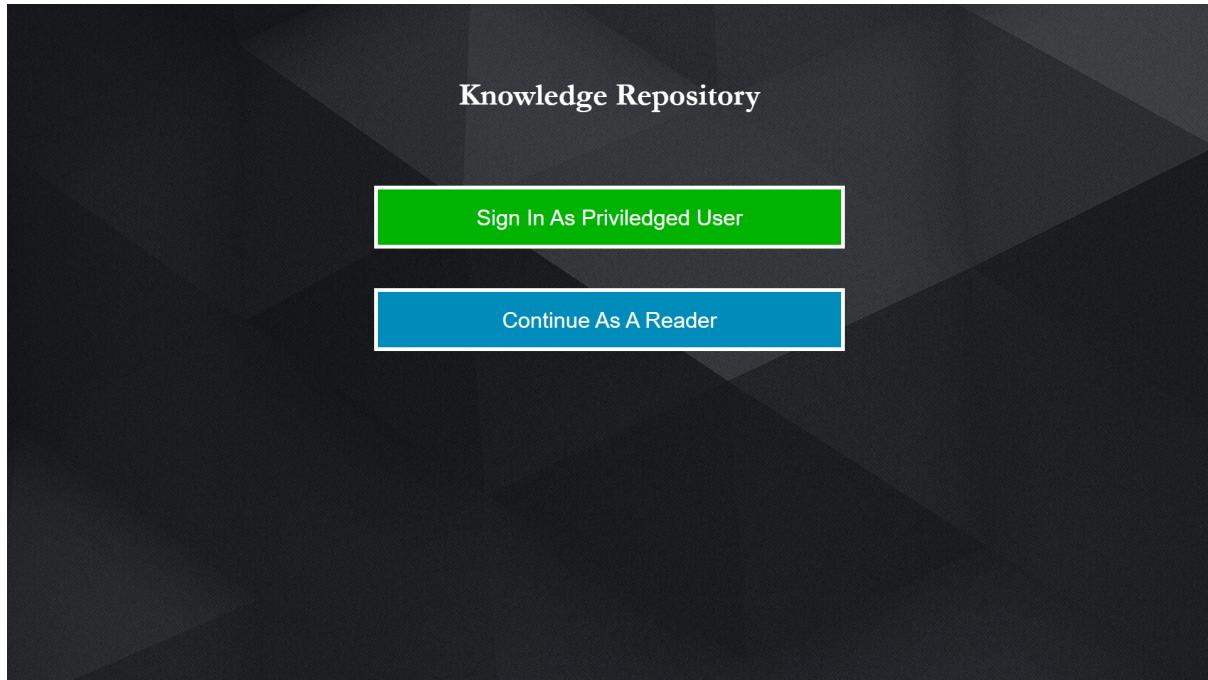


Figure 6.2.1

## Unauthorized / Anonymous Reader Home Page , Keyword based search

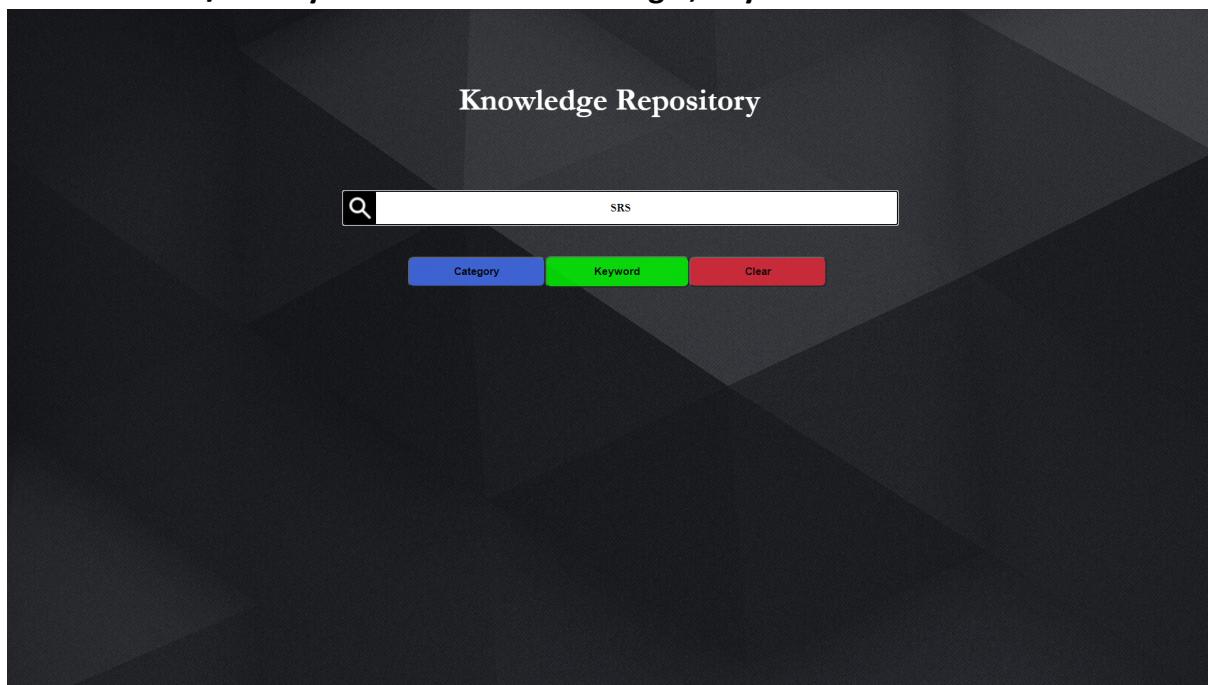
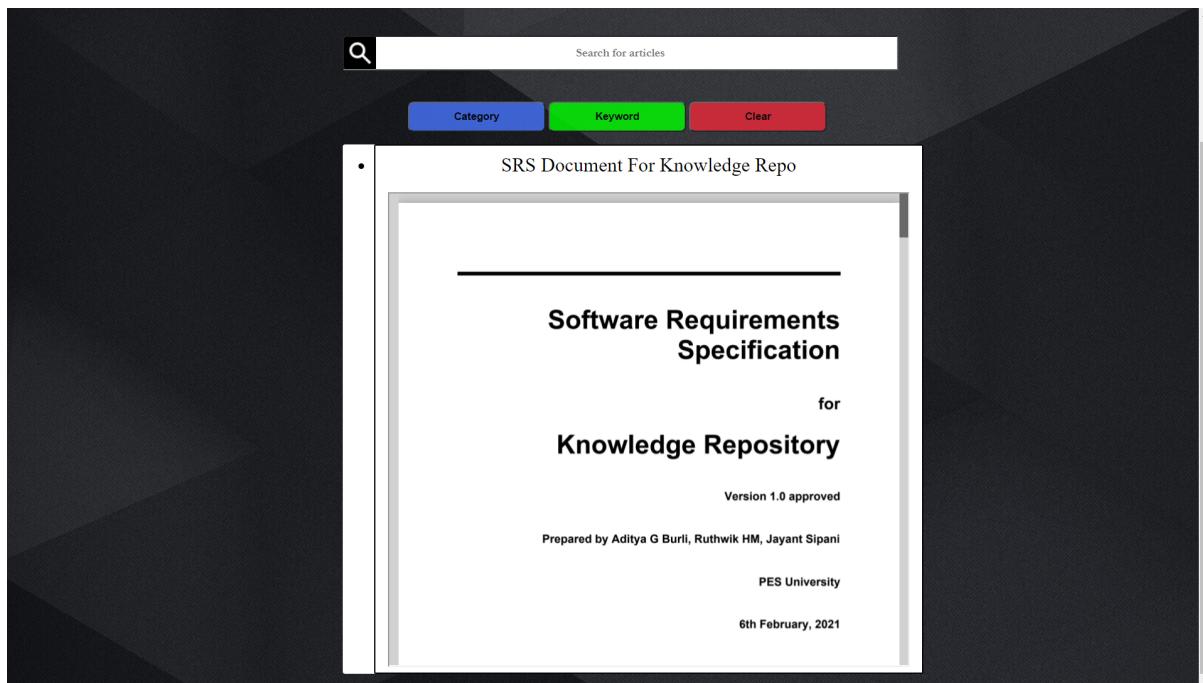


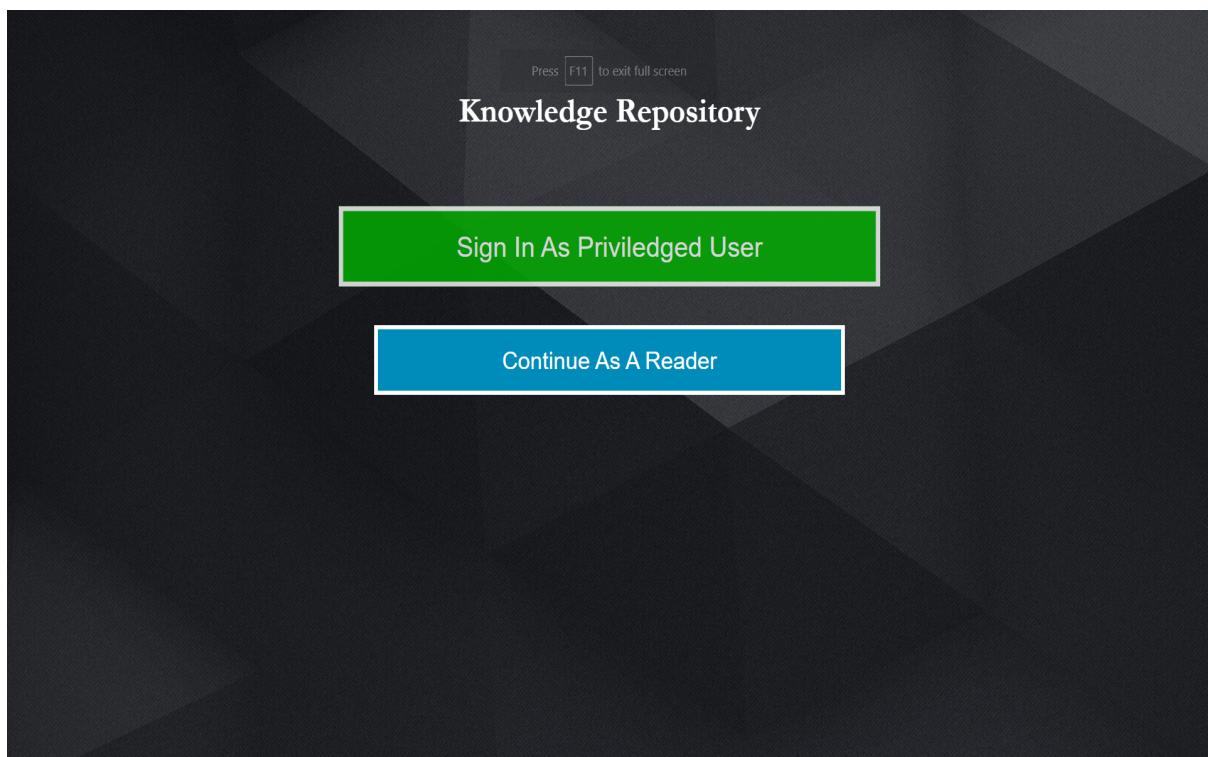
Figure 6.2.2

## Results of keyword based search , displayed in read only format for unauthorized users



**Figure 6.2.3**

**Authorization Page For Privileged Users (Downloading Articles Permitted)**



**Figure 6.2.4**

<Continued>

## User Authentication/Sign In successful for Reader/Subscriber

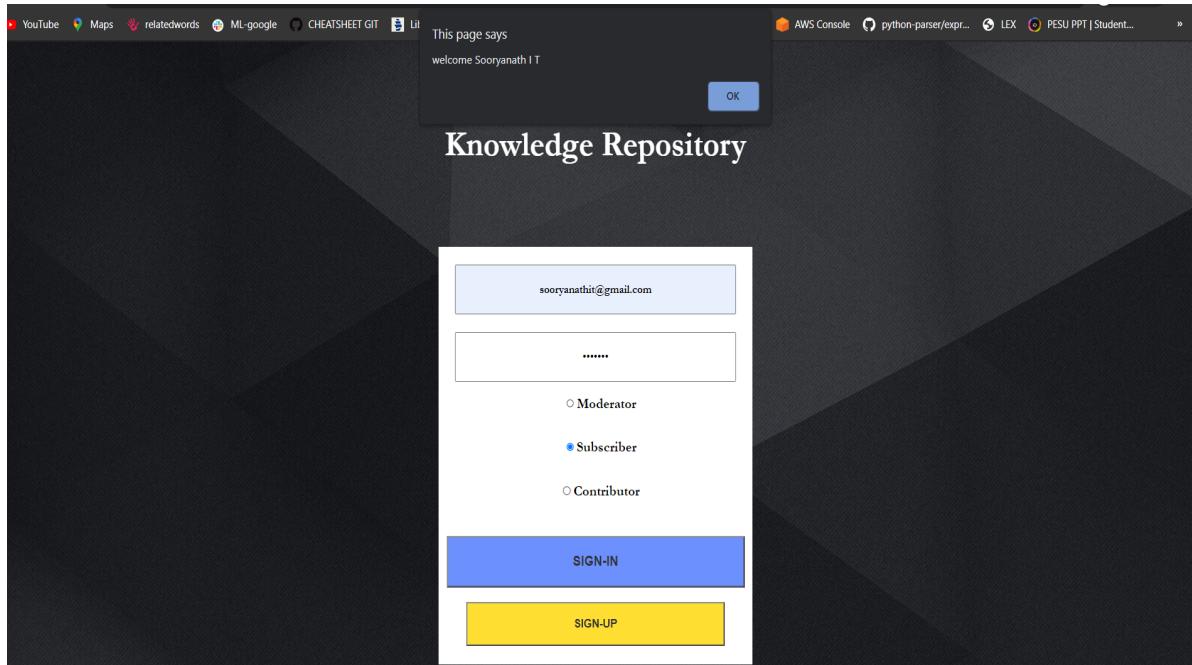


Figure 6.2.5

Similar UI for authenticated users as well , enter search phrase and search by keyword.

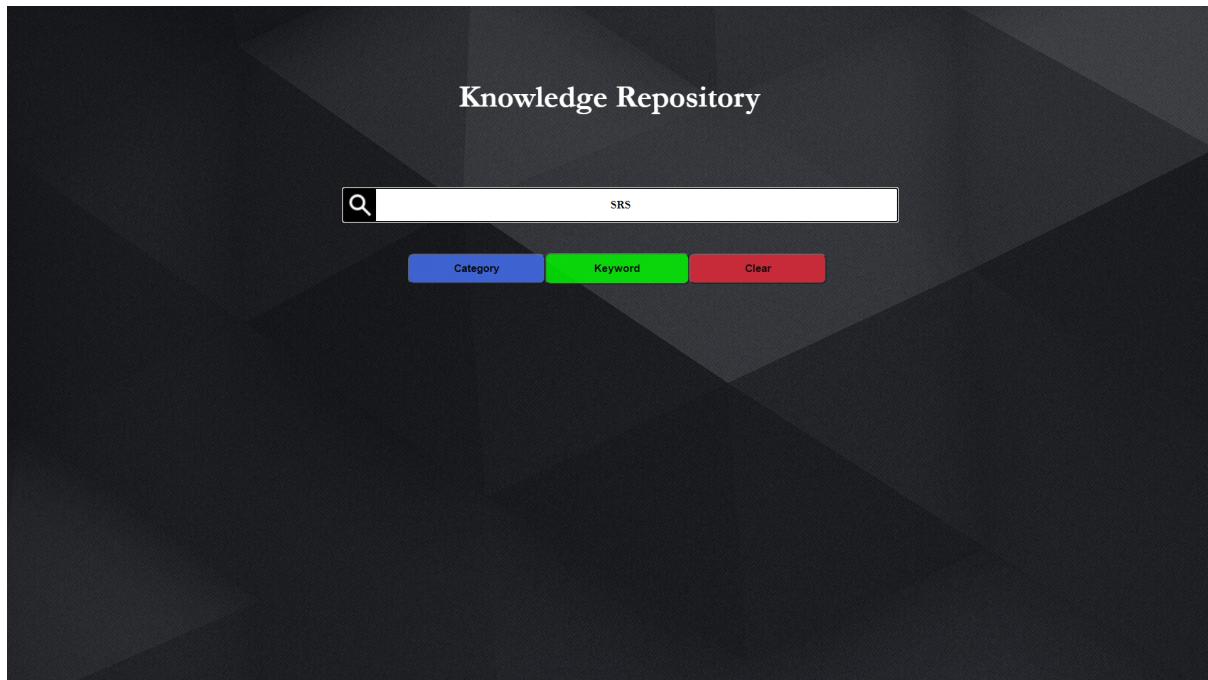
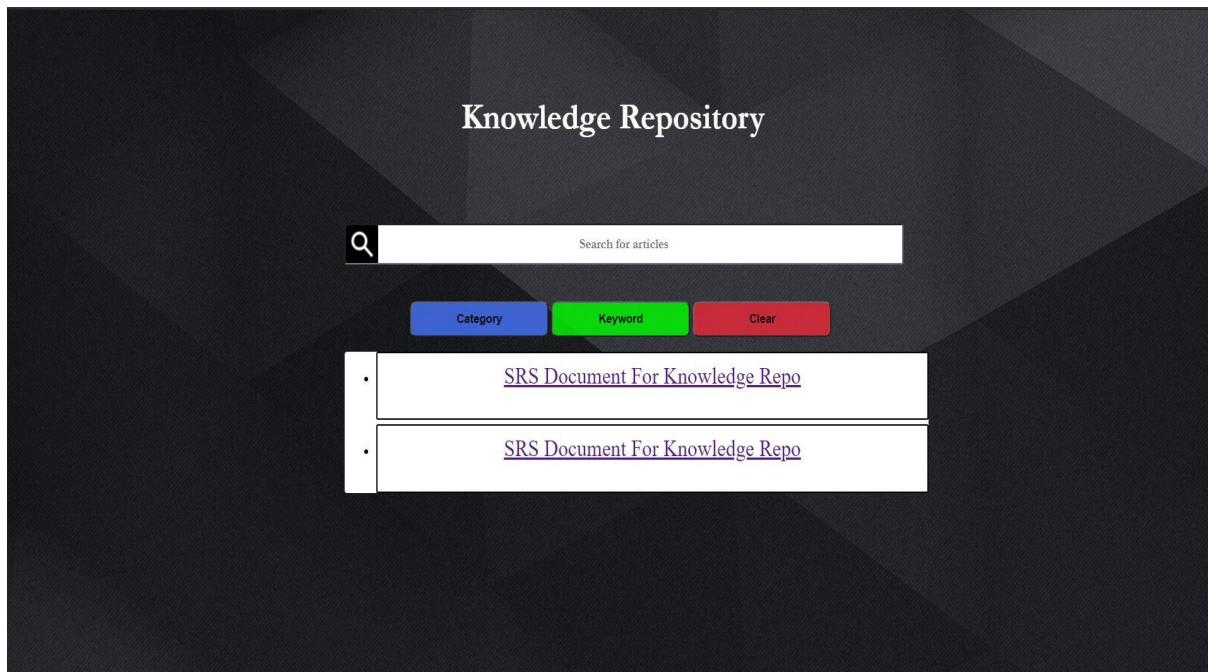


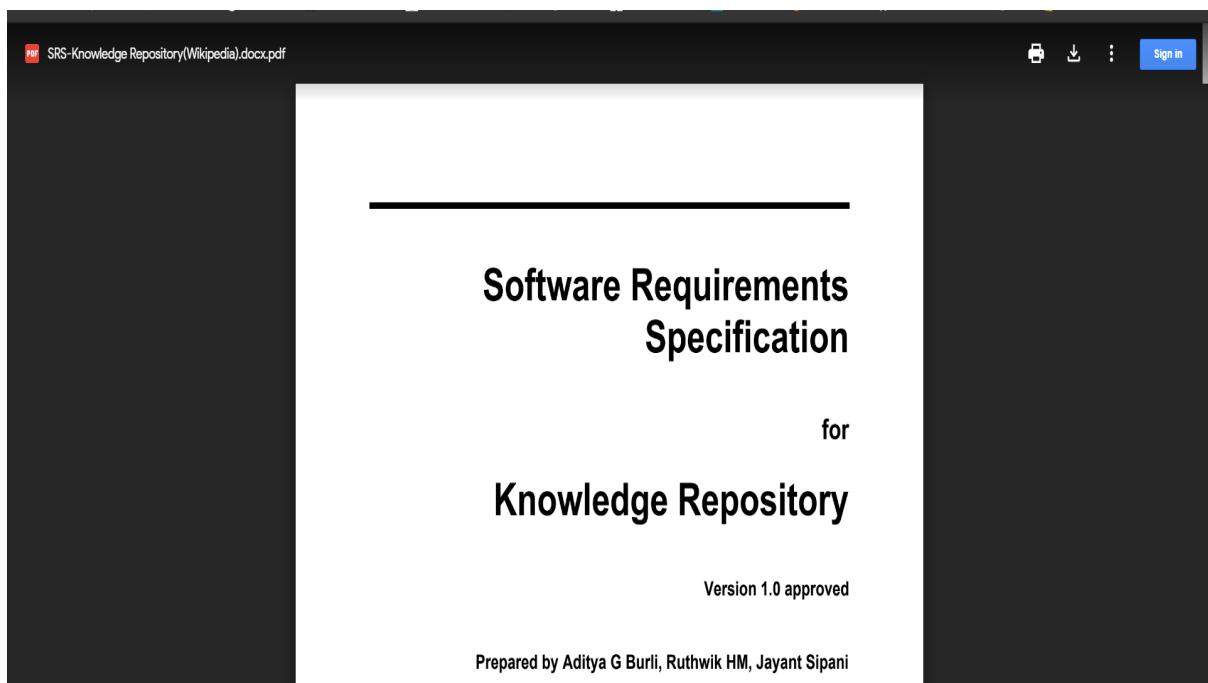
Figure 6.2.6

**Results displayed on the screen in form of link to the articles to read and download**



**Figure 6.2.7**

**Readable and Download option enabled article for authorized readers**



**Figure 6.2.8**

**Use case : Edit and Modify Articles.**

## Contributor Login

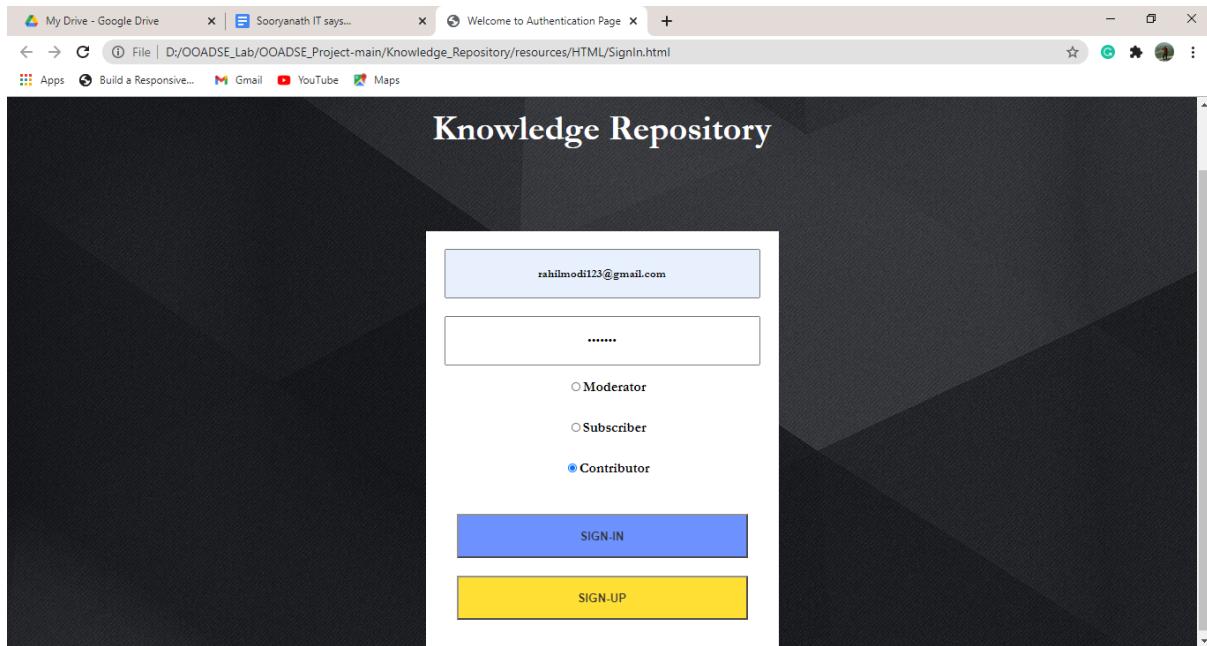


Figure 6.3.1

## Firebase initial configuration when no pending list

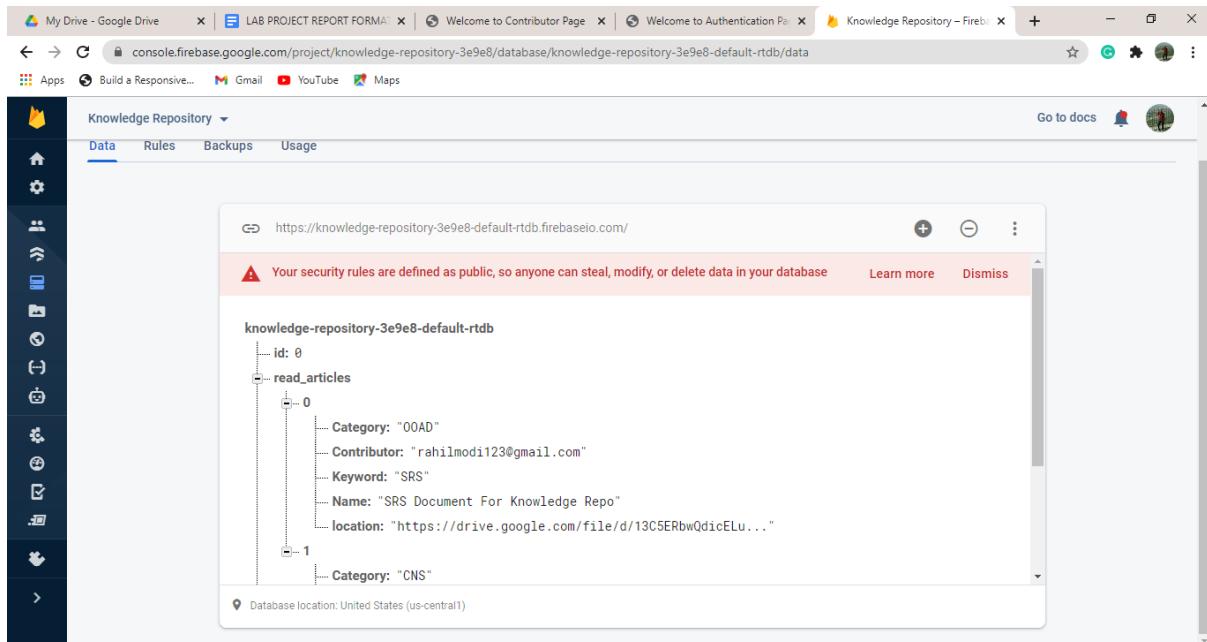
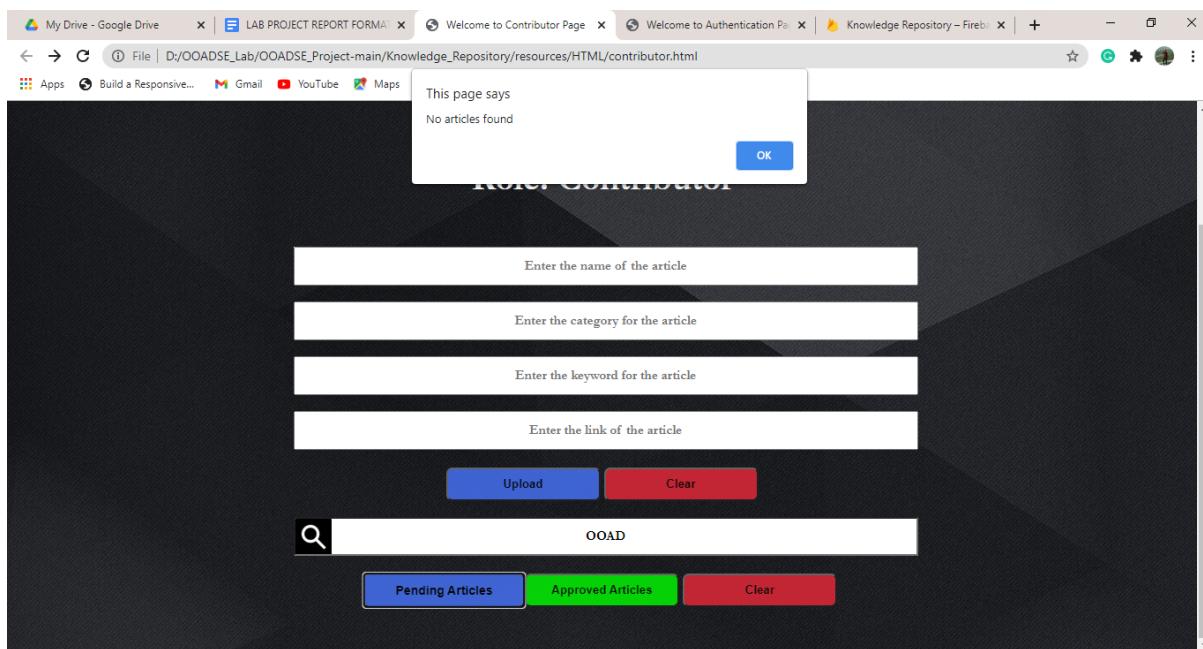


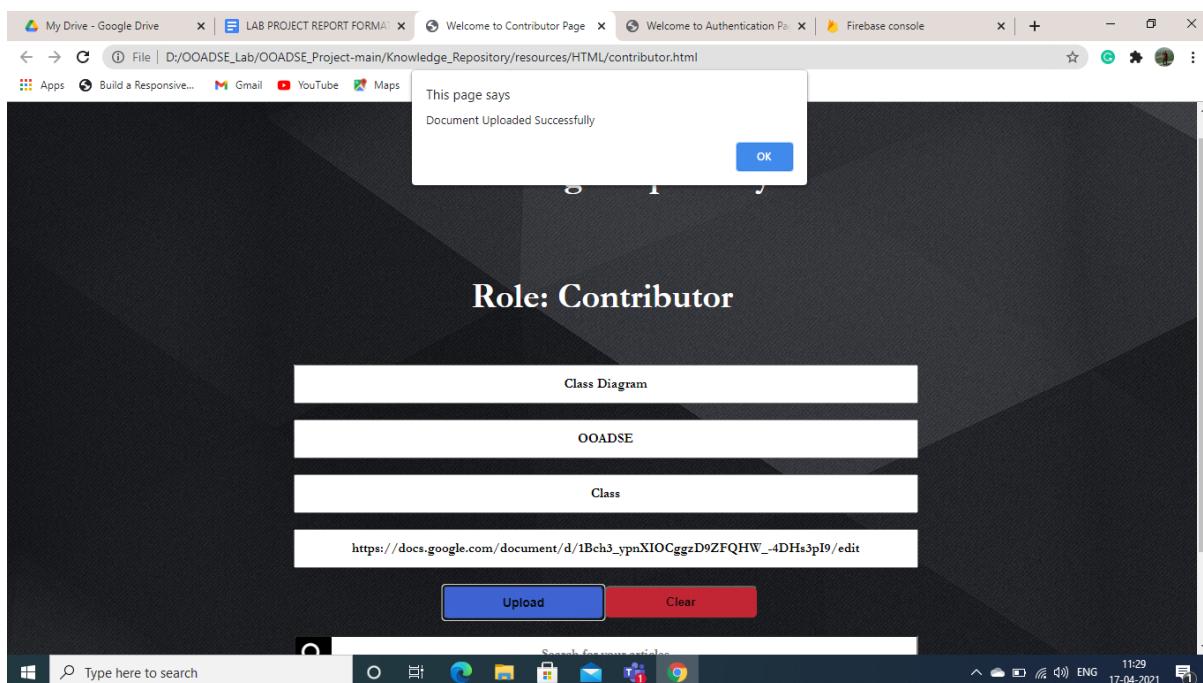
Figure 6.3.2

## Search for contributor's articles in pending list when no articles in pending list



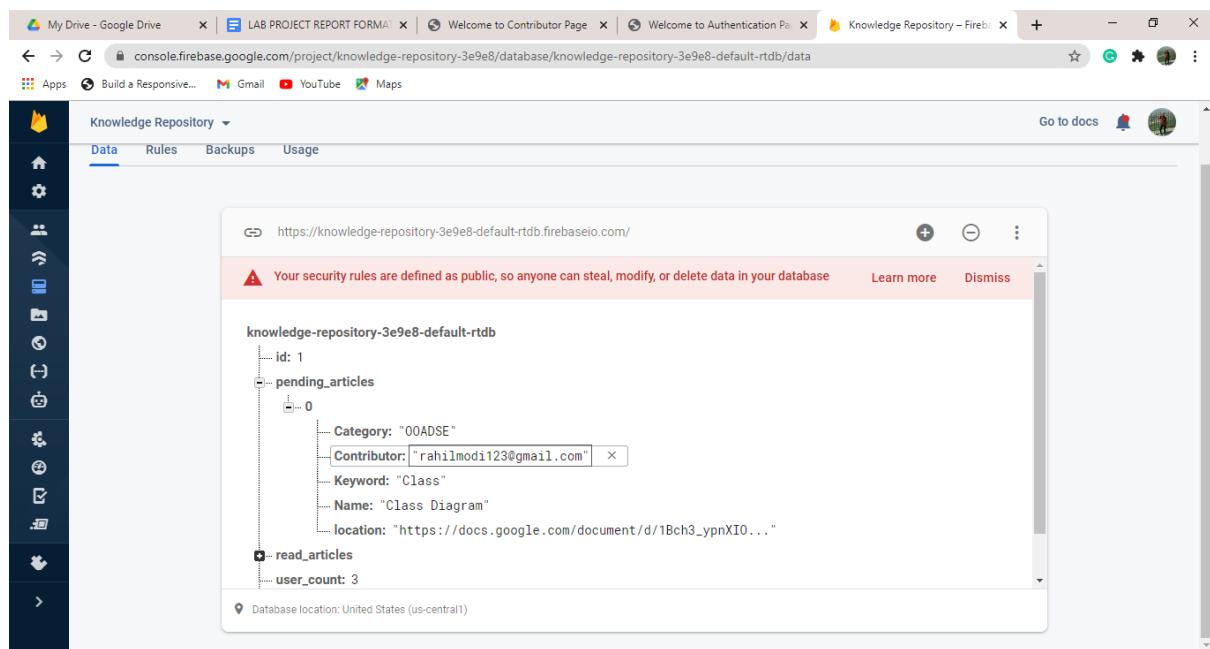
**Figure 6.3.3**

### Upload a document into pending articles list by the contributor:



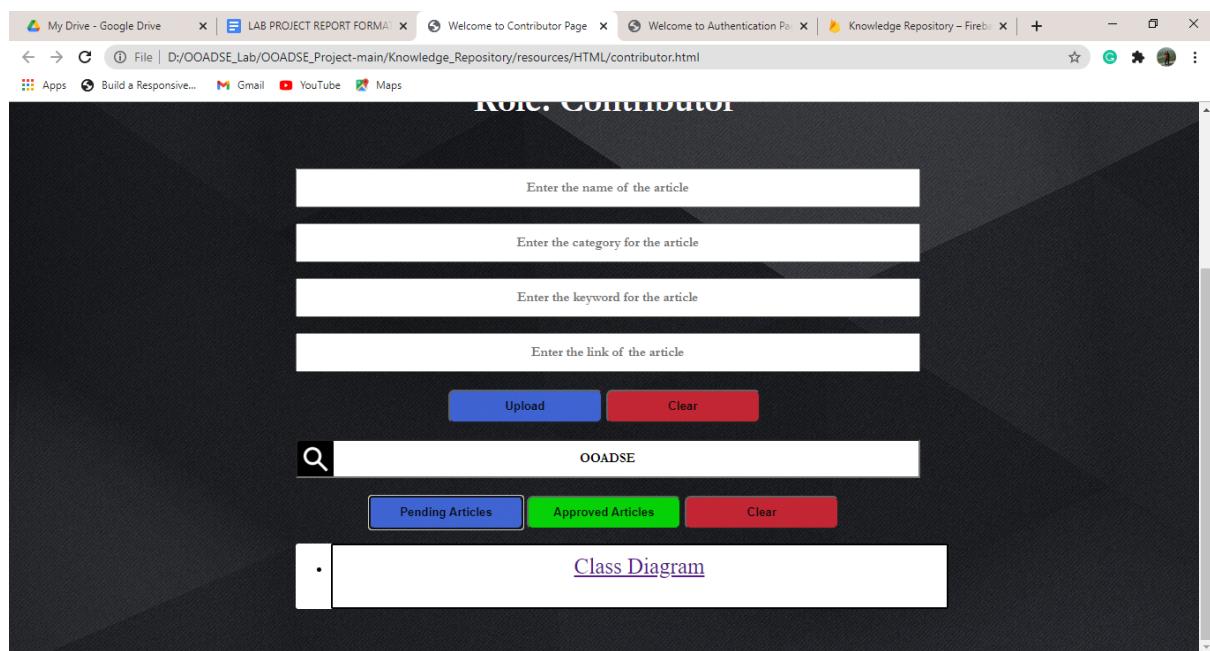
**Figure 6.3.4**

### Firebase configuration after uploading the document



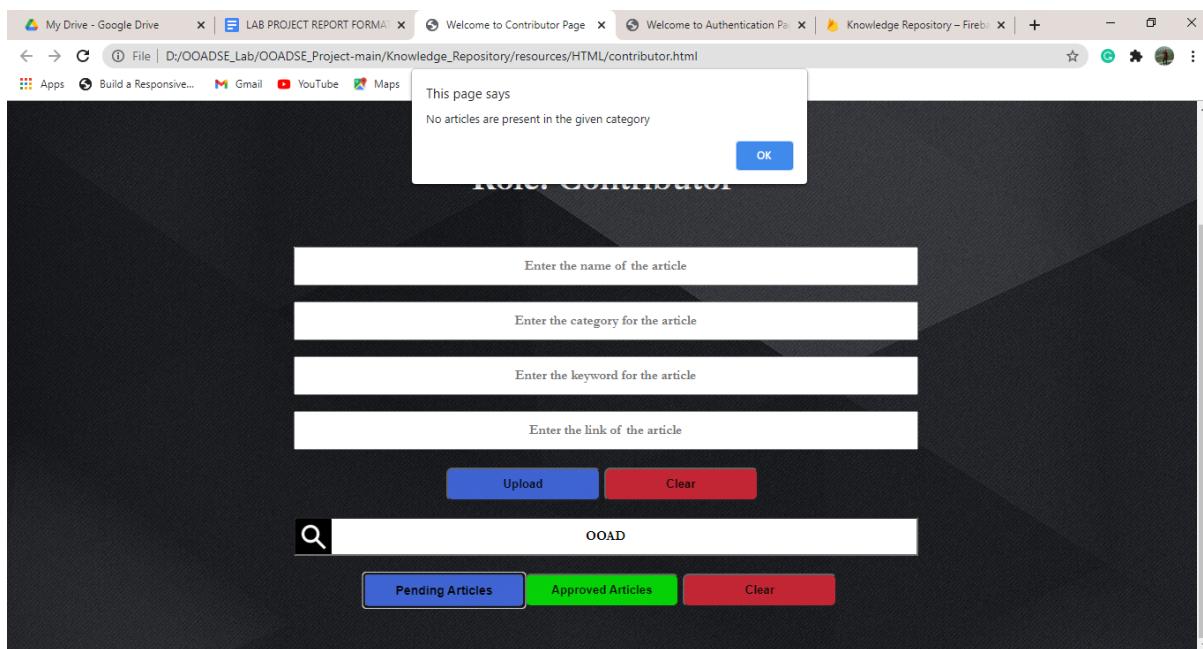
**Figure 6.3.5**

### Search for contributor's valid articles in pending list using category



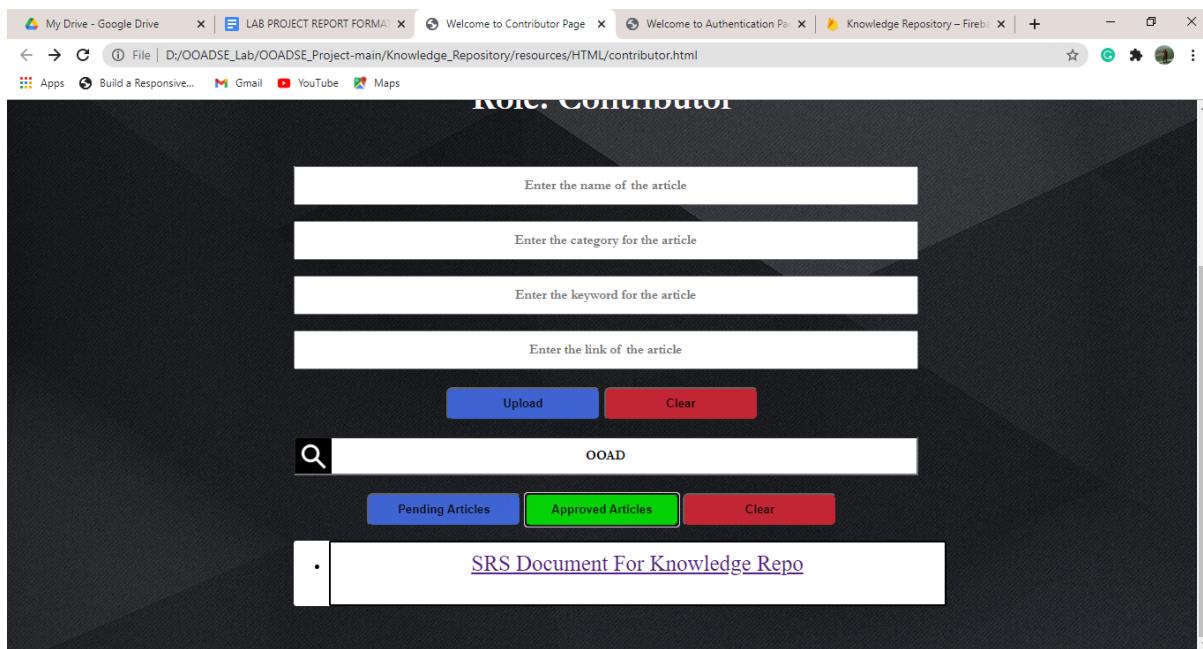
**Figure 6.3.6**

### Search for contributor's document not present in the pending list using category



**Figure 6.3.7**

### Search for contributor's document in reading list using category



**Figure 6.3.8**

### Use case : Approval and Review for content changes

## Moderator Login

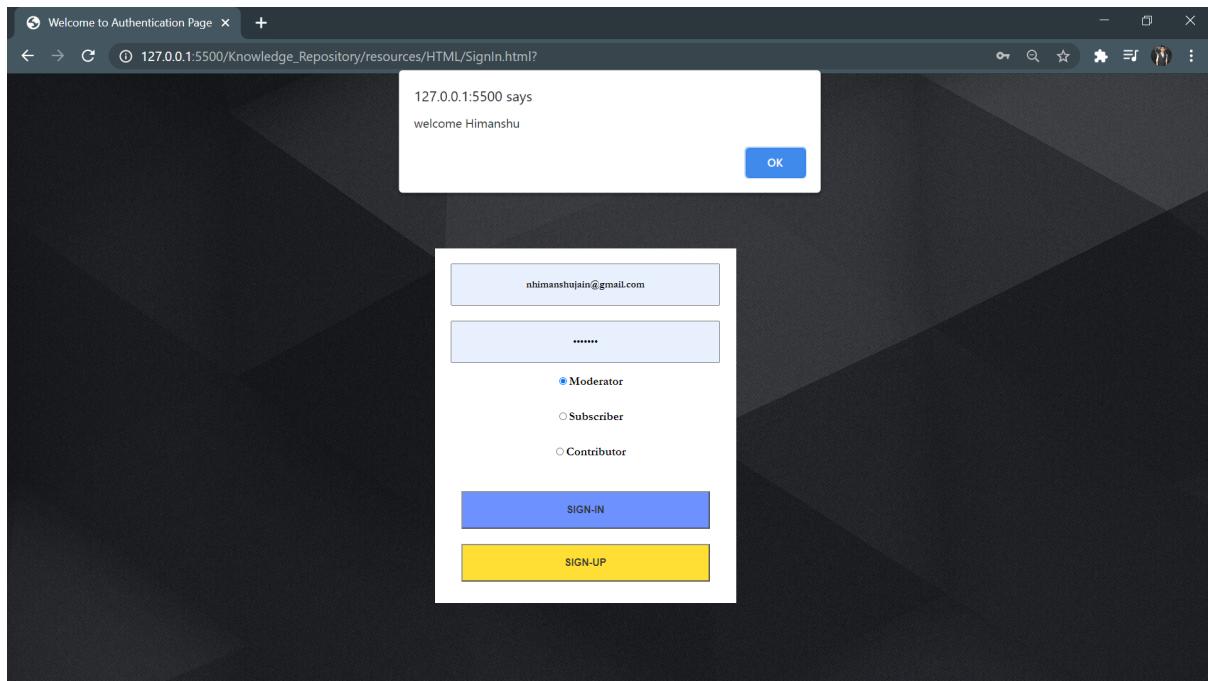


Figure 6.4.1

## List of pending articles to be approved

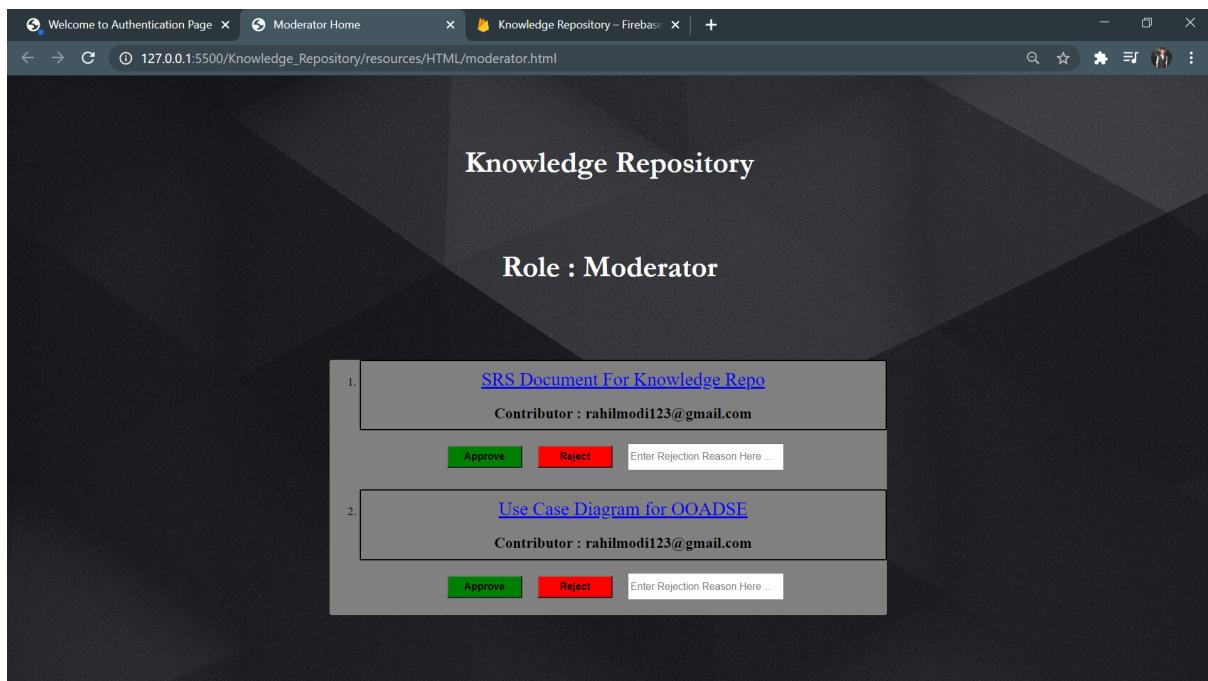


Figure 6.4.2

## Article Approved by the moderator and alert message showing the confirmation

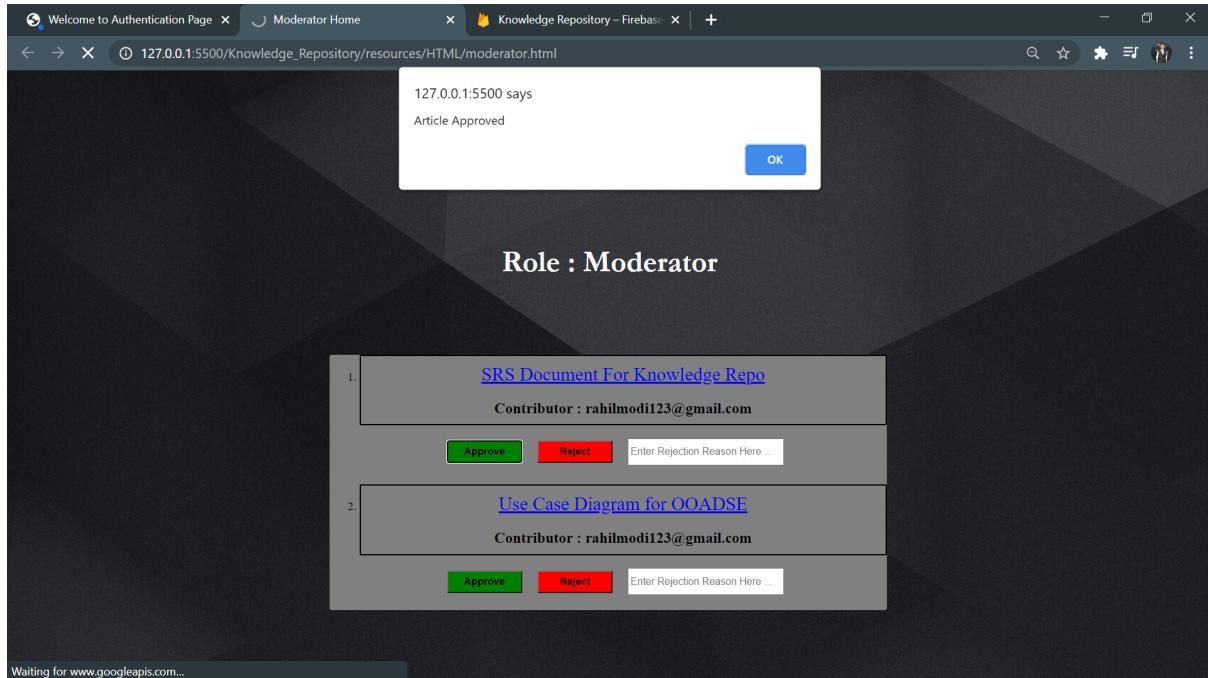


Figure 6.4.3

## Article Rejected by the moderator and the rejection message displayed in the alert

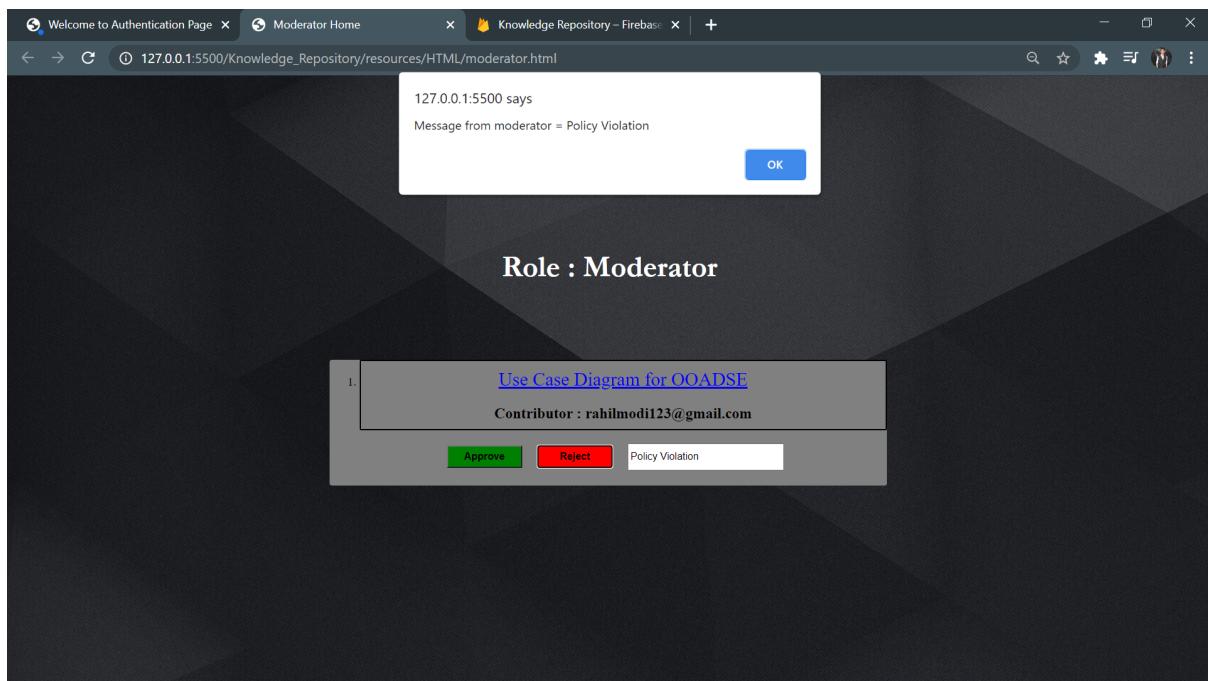
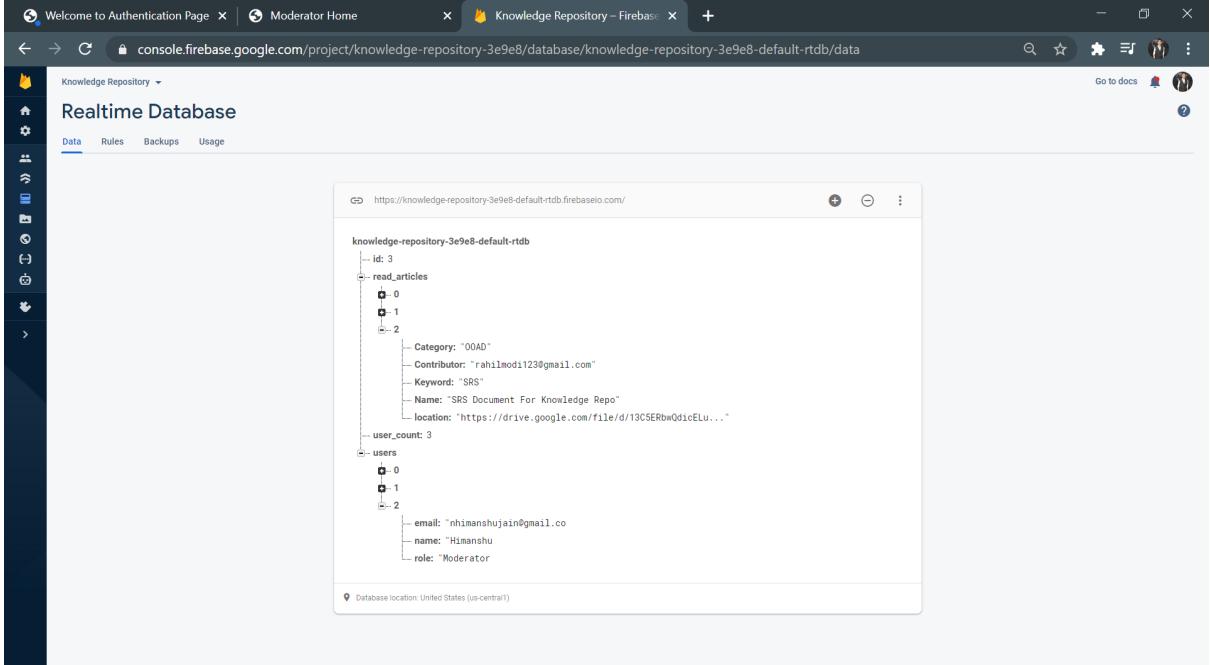


Figure 6.4.4

## Updated database after approval/rejection of articles



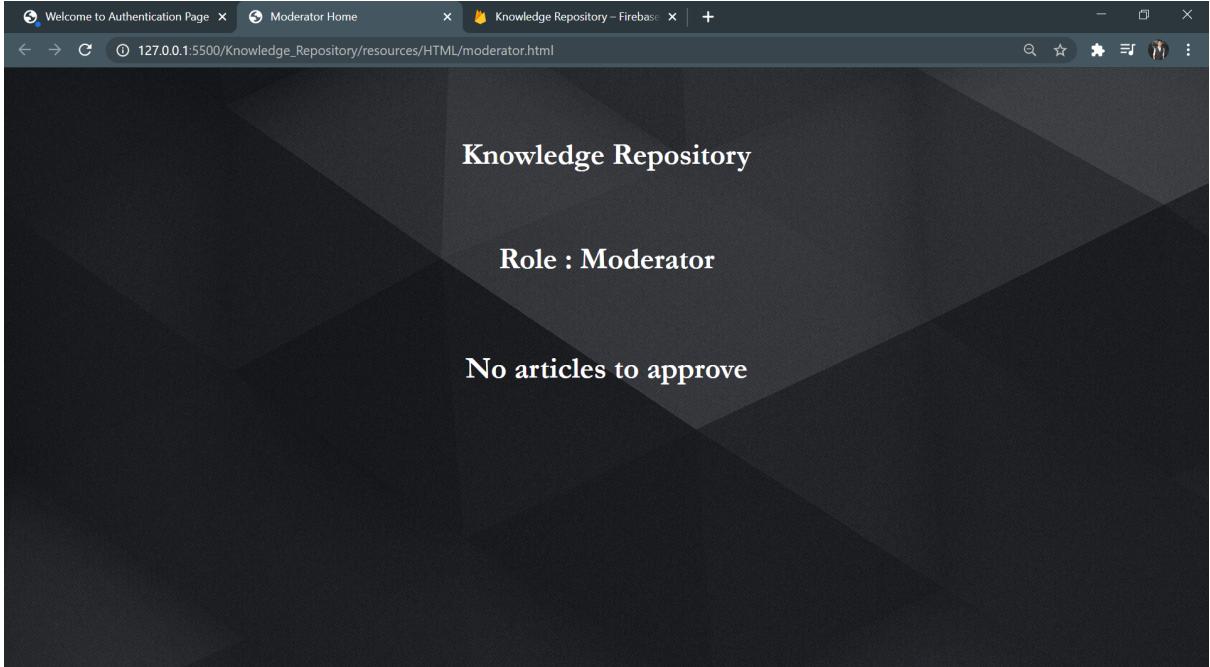
The screenshot shows the Firebase Realtime Database interface. The left sidebar has 'Knowledge Repository' selected under 'Realtime Database'. The main area displays a tree view of the database structure:

```
knowledge-repository-3e9e8-default-rtdb
  |- id: 3
  |- read_articles
    |- 0
    |- 1
    |- 2
      |   Category: "004D"
      |   Contributor: "rahilmodi123@gmail.com"
      |   Keyword: "SRS"
      |   Name: "SRS Document For Knowledge Repo"
      |   location: "https://drive.google.com/file/d/13C5ERbwQdicELU... "
  |- user_count: 3
  |- users
    |- 0
    |- 1
    |- 2
      |   email: "nhimanshujain@gmail.co"
      |   name: "Himanshu"
      |   role: "Moderator"
```

At the bottom, it says 'Database location: United States (us-central)'.

Figure 6.4.5

## No pending articles to be approved by the moderator



The screenshot shows a dark-themed web page titled 'Knowledge Repository'. The top navigation bar includes 'Welcome to Authentication Page', 'Moderator Home', and 'Knowledge Repository – Firebase'. The main content area displays the following text:

Role : Moderator

No articles to approve

Figure 6.4.6