1.INTRODUCTION

* 1. Overview of the project

There are lots of search engines available, but for results based on biological only domains, there are none.

Our application will provide conclusions for this particular domain only.

This application will take some mechanical or biological terms in the search bar, and return filtered outputs based on mechanical and natural correspondence.

* 1. Problem Definition

The relationship between physical laws governing motion and mechanics of artificially built objects and their natural counterparts found in the environment is of great interest in the field of biotechnology.

The goal is to design and develop a system that allows you to study the natural analogies related with physical laws of mechanics.

2.PROPOSED SYSTEM

2.1 Description of Target Users

This application is built so that folks from both mechanical and biotech department can use it for their research, analysis, examination, investigation, and exploration purposes.

This application can also be used by people outside of the domain to infer biological knowledge of these mechanical and physical attributes.

2.2 Description of Proposed system.

The user shall enter his terms to be searched on a text area.

We can provide filters based on options “land”, ”air” and ”water”.

The results will be displayed by sorting based on relevance.

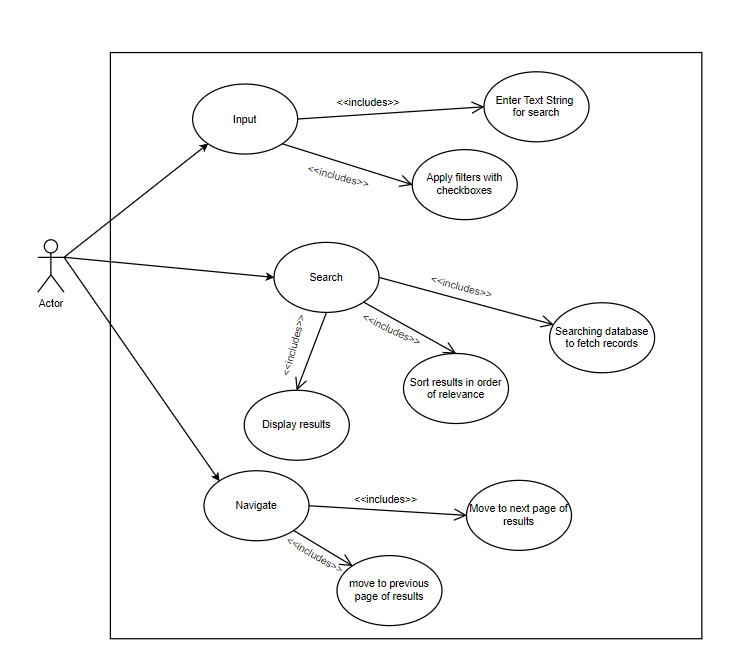
Along with a short description, book and author name, and the page number where we can refer that term in the book.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Requirement Specifications  
 3.1.1 Functional Requirements

* Show biological analogies related with input text given containing physical laws
* Allow user to filter search based on Land, Air, Water.
* Display the result, such that the most relevant result is shown first.
* Display separate fields of results like titles, content related author and book etc.
* Allow multiple searches, so the program behaves as a search engine.

3.1.2 Use Case Diagrams



3.1.3 Use Case Description

3.1.3.1 Input

* Actor: System user
* Trigger: User starts writing in the text field or checks on of the checkboxes.
* Pre-conditions: system is in running condition.
* Action: search the database for relevant records if search is performed following the input.
* Exception; system failure
* Outcome: system is ready condition to search and has all the info required to search in database.

3.1.3.2 Search

* Actor: System user
* Trigger: User presses the search button, provided beside text entry field
* Pre-condition: System is in running condition.
* Action: Search the database and Sort the results found in order of their relevance, display the results.
* Exceptions: System Failure
* Outcome: Results are displayed to the user alongside a count of how many documents were found.

3.1.3.3 Navigate

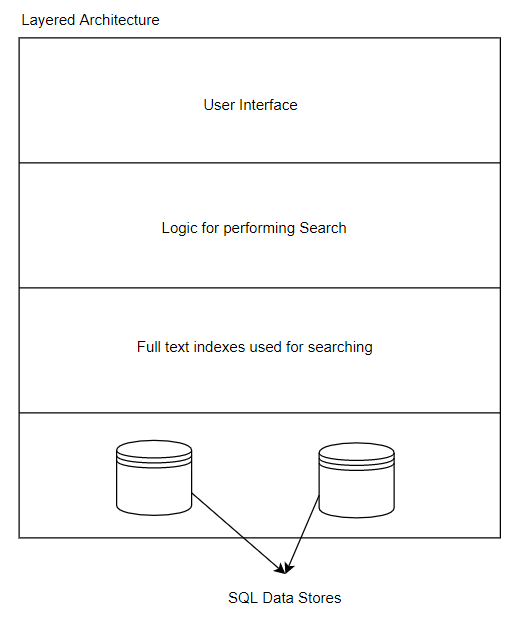
* Actor: user
* Trigger: user presses either of the next or previous button
* Action: if further results are present that can be displayed, displayed results are changed to display other records.
* Pre-condition: system is in running condition
* Exception: system failure
* Output: when next or previous records will be displayed as asked.

3.1.4 Non Functional Requirements

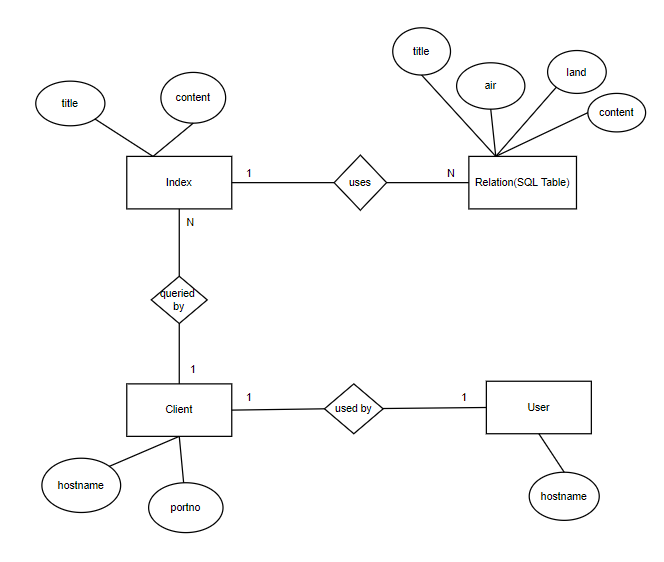
* Results should be displayed pretty much instantly
* Results shown should be navigable (if many records are found). Seamlessly (instantly).
* System should use resources effectively.

4.SYSTEM DESIGN

4.1 System Architecture



4.2 ER Diagram



4.3 Dataset description

We used 3 books,  
Animal Locomotion Physical Principles and Adaptations.

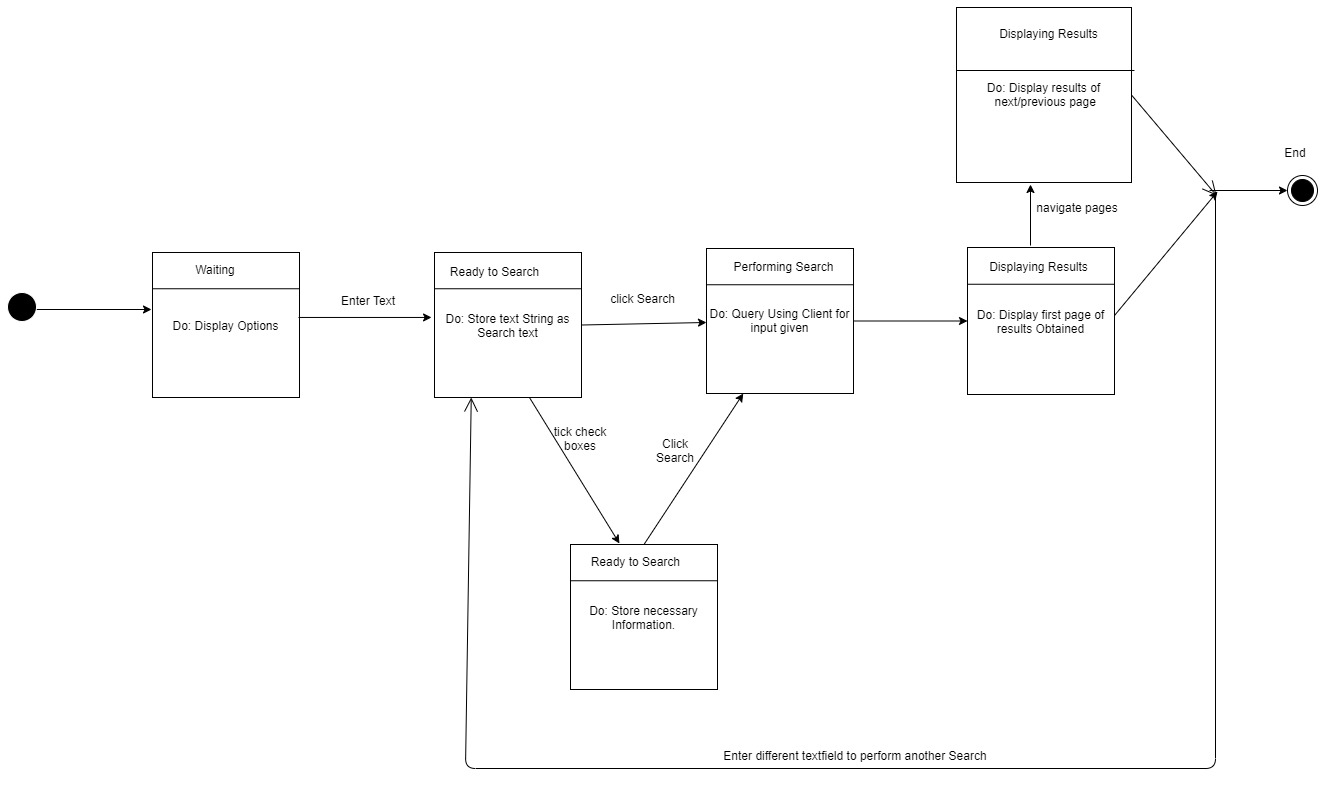
History effects, oxidative stress, and the evolution and expression of animal signals.

Adaptive motion of animals and machines.

Manually we selected the keywords and description for each of those terms.

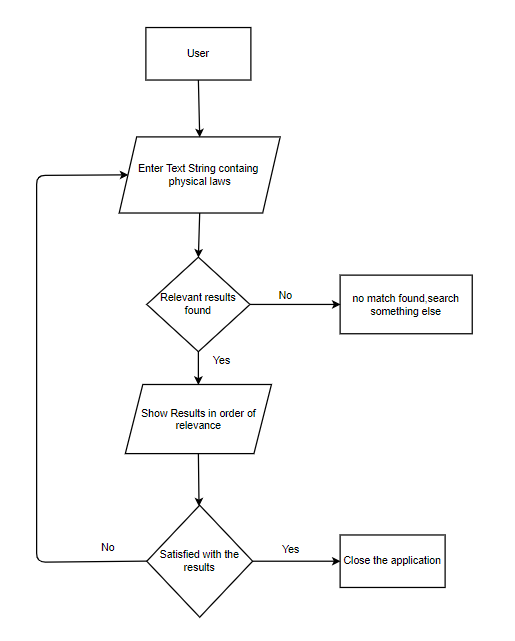
This shall be presented along with the book’s name and page number where that term is found.

4.4 Stat Transition Diagram



5 IMPLEMENTATION

5.1 Proposed Methodology (Flow Chart)



5.2 Description of Modules

5.2.1 Indexing

Sphinx tool is used to build indexes out of relational tables present in the SQL database

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Air | Land | Water | kwrds | title | content | … |

The encircled columns are used by indexer to perform full-text indexing of the texts present in those columns.

The resulting index has the remaining columns as attributes . we can query this Index directly and all the attributes will be returned for the indexes matched.

Input : Sql Relation

Output : Full-text Index

5.2.2 Searching

To perform searches on the build indexes we can query the indexes directly following either the SQL-like syntax if we query on the port 9306, queries can also be made on port 9312 using the Client object available as part of Sphinx API.

Pseudo-code :

For queries on port 9306

Select <attribute> from <indexname> where match(‘match\_string’);

For queries on port 9312

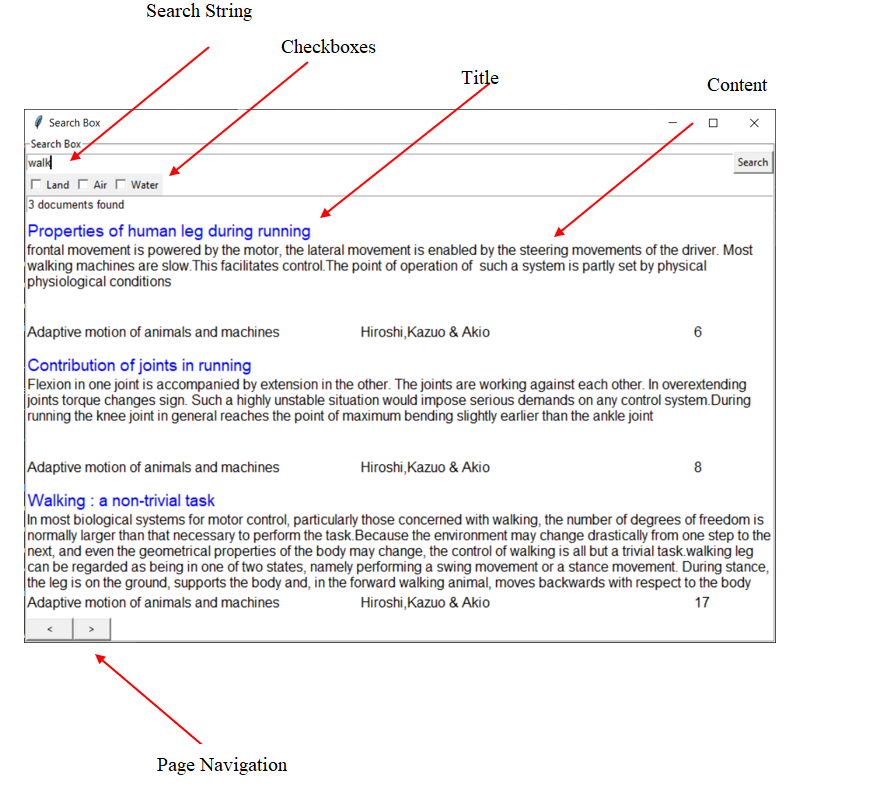
client = Sp.SphinxClient(‘hostname’,portno);

client.Query(‘match\_string);

Input : Query String

Output : Documents found in order of relevance

5.2.3 User Interface

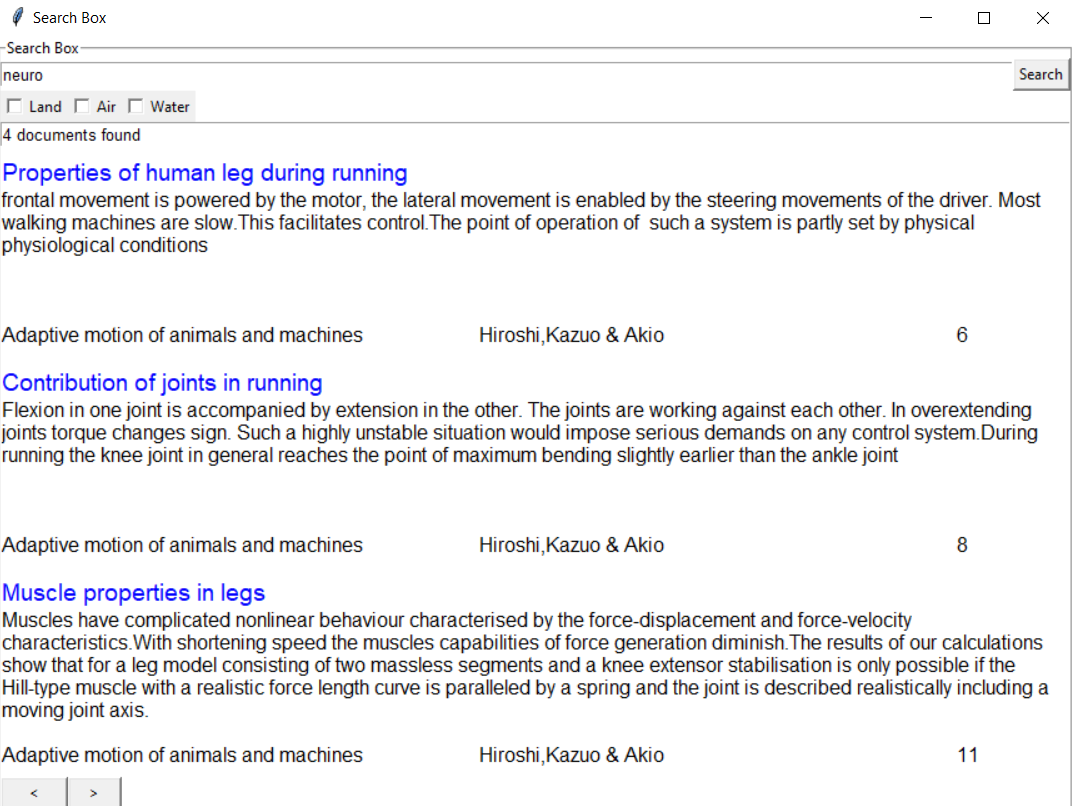


Input : Press Search Button  
Output : Documents Displayed

6.1 Testing and Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement id** | **Test id** | **Input description** | **Expected output** |
| **1.1** | **1.1.1** | **Entering the string “runned” in search bar** | **Displays biological terms related to the word run** |
|  | **1.1.2** | **Entering the string “running” in search bar** | **Displays biological terms related to the word running, run** |
| **1.2** | **1.2.1** | **Enter the string “walk”** | **Display results related to walking running, legs etc.** |
| **2.1** | **2.3.1** | **Entering the string propulsion in search bar, with land filter on.** | **Displays biological terms related to the word propulsion, and sort out results that are based on only land** |

7.RESULTS AND DISCUSSION



We see that the system is able to recognize exact matches for keywords as well as partial matches, also different word forms.  
The results returned by the system are sorted in the order of relevance(weightage based on matches)

Combination of words are joined using logical OR’s to make up search strings.

This can be lifted to logical AND’s if our database is big enough.

8. CONCLUSION AND FUTURE SCOPE

There are not many such implementations in this domain. This model, can be deployed online as a search engine.

We can also add images as a search result.

The database can be hosted online, there are various hosting providers offering MariaDB.

The text displayed could be hyperlinks which when clicked will open up results in your default browser.