WEB TOOL FOR NEWS SUMMARIZATION

Software Project lab-3

NOVEMBER 25, 2018

INSTITUTE OF INFORMATION TECHNOLOGY
UNIVERSITY OF DHAKA

WEB TOOL FOR NEWS SUMMARIZATION

Submitted by

Fazle Rabbi

BSSE-0725

Supervised by

Dr. Zerina Begum

Professor

Institute of Information Technology

University of Dhaka

Letter of Transmittal

November 25, 2018
BSSE 4th Year Exam Committee
Institute of Information Technology
University of Dhaka

Dear Sir,
I have prepared the report on my project “Web Tool for News Summarization”. This report includes the details of each steps I followed to collect the requirements.

The primary purpose of this report is to summarize my findings from the work that I completed.

Sincerely yours,

Fazle Rabbi
BSSE 0725

Document Authentication

This project document has been approved by the following persons.

Prepared by

Fazle Rabbi

BSSE-0725

Approved by

Dr. Zerina Begum

Professor

Institute of Information Technology

University of Dhaka

Letter of Endorsement

November 25, 2018

To Whom It May Concern

Subject: Approval of the report

This letter is to clarify that all the information mentioned in this document is true. The project mentioned here have had successful involvement of Fazle Rabbi, BSSE 0725 from Institute of Information Technology, University of Dhaka.

I wish him all the best and hope that he will lead a successful career.

Project Supervisor

Dr. Zerina Begum

Professor

Institute of Information Technology

University of Dhaka

Acknowledgement

I am happy that I have been able to complete this analysis as well as the report for my project “Web Tool for News Summarization” by the grace of Almighty. Undoubtedly it is a hard task to complete designing and analyzing requirements, implementing and testing software perfectly and then preparing a report on it. I am also very thankful and grateful to my respected supervisor Dr. Zerina Begum. It could be more difficult task for me if she would not share her valuable knowledge and lead us to complete whole procedure wonderfully.

Abstract

Related news of a specific topic are crucial need to analyze a lot in shorter time. Often the summarization is also demandable for generating of birds view of documents. Searching time would be reduced for a desired news. At present there are few web summarizer tools are existed like autosummarizer, resoomer etc which can reduce the size of a passage by summarizing. But those tools are not for summarizing news related to a topic. So a web tool has been developed which can provide summarized news articles related to a topic.Before summarization, news articles for daily newspaper are collected. When the news articles are collected, the server is ready for taking search topic. Lexrank summarization technique is used for both single document and multiple document summarization. After entering the website summarized news for a given topic can be fetched. It is found that the summarized news article size is smaller than the original news content. Also the summarized form conveys the main contents of news articles.

Table of Contents

Table of Figures

1. Introduction

This chapter describes the objectives of this report as well as the audiences who should have to go through this report for individual purposes.

1.1. Purpose

This document is based on the technical documents consists of Short version of Software Requirements Specification (SRS) and Design Modeling for the “Web Tool for Summarizing News Articles” proposed to be developed in Software Project Lab 3. It includes all necessary requirements to develop this application no matter whether they are functional or non-functional. The information about the requirements here have been organized systematically so that everyone can easily figure out a summarized concept about The Web Tool.

1.2. Intended Audience

This SRS is intended for several audiences including customers, designers, developers and testers.

- The customers will use this document to ensure their functional requirements has properly been met in the end product.
- The project manager will use this document to plan a schedule and estimate cost and delivery date.
- The designer will use the high level architecture outlined in this document to design the complete system to meet the customer requirements, which is also detailed in this document.
- The developers will periodically refer back to this document to ensure their implementation corresponds to the customer requirements.
- The testers will use this document to get a clear picture of functionality of individual components and whole system to test against.

2. Project Description

In the Software Project Lab(SPL)-3, a web tool will be implemented where a passage from news/article/essay can be summarized using automated text summarization technique. The web tool will take any topic and url from a user, find related passage from the url and finally present summarized text to the user. Some text summarization techniques will be implemented one by one and finally the best method will be used in server. The summarization will be applicable for English natural language.

2.1. Quality Function Deployment(QFD)

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to my project, the following requirements are identified by a QFD.

Normal Requirements:

- A User can search a news topic with a keyword.
- Web server will present summarized news based on the keyword with published date.
- News will be collected from daily newspapers (prothom-alo, bdnews24, bbc-bangla, ittefaq).
- News summarization will be applicable only for english newspapers.
- User gets news in shortest possible time.

Expected Requirements:

- Provided keyword will be matched with newspapers headlines.
- Summarized news should contain the main theme of a news.
- News will be summarized by extractive approaches (most important and related sentences will be fetched as summarized news)

Excited Requirements:

- News will be summarized by abstractive approach (Fully new sentences will be generated from the news.)

2.2. Scenario

A user can access the tool with a web browser. The server activities can be divided into three major parts:

Collect News:

A News Collector tool will collect news from daily newspapers automatically in a consecutive period of time and save all collected news articles to files. The below list of newspapers will be used for collecting news:

- https://www.thedailystar.net
- http://www.observerbd.com
- https://bdnews24.com
- https://en.prothomalo.com
- https://www.clickittefaq.com
- http://english.kalerkantho.com

Show News:

A user can access the tool with a web browser. After entering, a search bar will be appeared. The user can search a news topic by entering with a keyword in the search bar. After clicking search button the web tool will present summarized news articles from newspapers based on the searched topic. The final output will be summarized text of news articles with published date, title, url link and newspaper name sorted by published date in descending order.

Summarize News:

An automatic summarizer tool will be developed in the server. Using the summarizer tool the news articles will be summarized. The summarizer tool will summarize a whole news into some shorter important and related sentences

3. Scenario Based Modeling

Scenario based modelling is an inexpensive rapid prototyping technique. This method is effective when systems are being built with the requirements vaguely known at the outset. Users are involved right from the start, to build prototypes evolving towards the final product. The users are also involved with the testing of the prototypes which is essential for the validation of requirements and help the users to gain an initial experience of the final system during the development itself. This method involves techniques which are applied by one or more professionals working alongside users who are expected to provide and specify their requirements at the beginning as well as evaluate and approve the system upon completion. The user (in a passive capacity) and the designer/builder (an active partner) cooperate to reach a working model where the means of communications are by the examination of preliminary models such as the initial narratives, paper models and graphical representations built to represent the final system functions.

3.1. Use Case Diagrams

A use case diagram is a graphic depiction of the interactions among the elements of a system. The purposes of use case diagrams are:

- Gathering requirements of a system.
- Getting an outside view of a system.
- Identifying external and internal factors influencing the system.
- Showing the interactions among actors.

The first step in writing a use case is to define the set of actors that will be involved in the story. Actors are of two types. They are:

- Primary Actors: Primary actors are the actors using the system to achieve a goal. They both consume data and produce information.
- Secondary Actors: Secondary actors are the actors that the system needs assistance from to achieve the primary actor's goal. They either consume data or produce information.

Once actors have been identified, use cases can be developed.

Level 0: Present Summarized News
Primary actor: User, News Presenter.
Secondary actor: News Collector, Summarizer
Goal in context: The diagram refers to the overview of the web tool for news summarization.

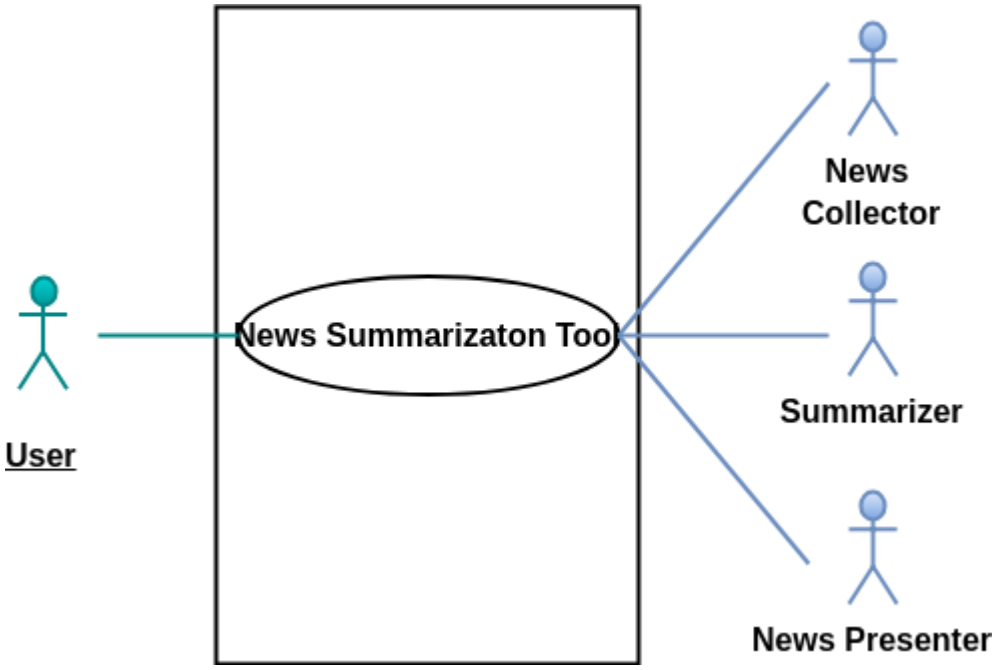

Figure 1: Level 0 of Use Case

Level 1: Present Summarized News
Primary actor: User, News Presenter.
Secondary actor: News Collector, Summarizer
Goal in context: The diagram refers to the details of the web tool for news summarization.

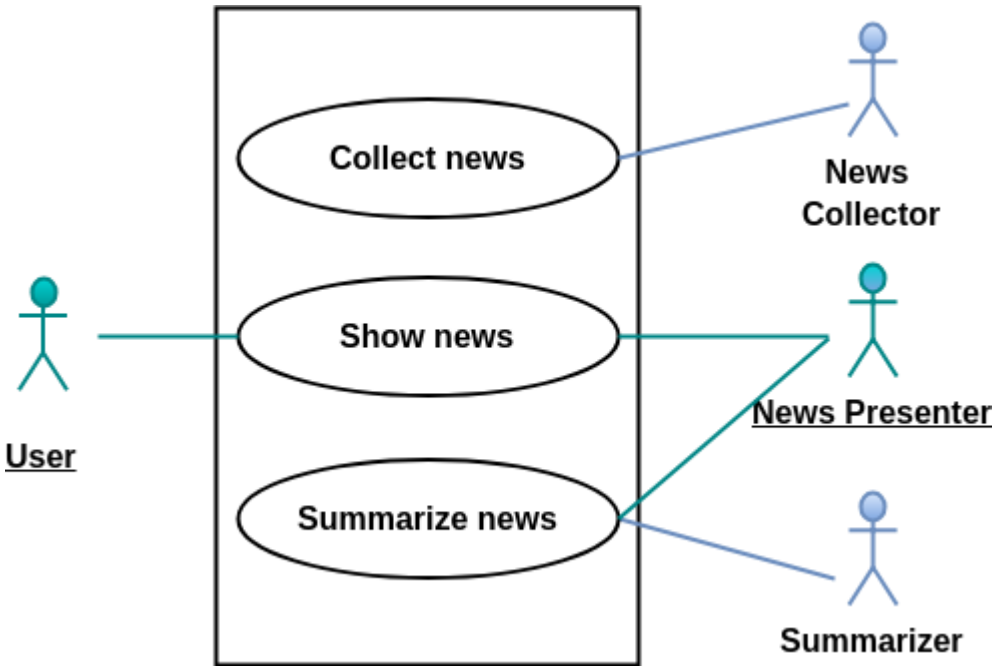

Figure 2: Level 1 of Use Case

Level 1.1: Collect News
Secondary actor: News Collector
Goal in context: The diagram refers to the details of the news collection module of level 1.

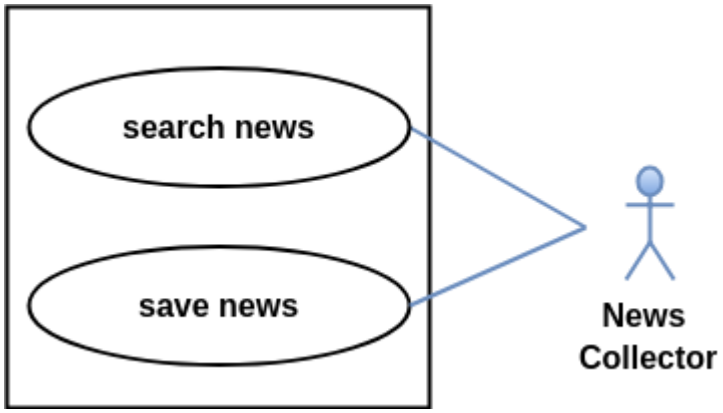

Figure 3: Level 1.1 of Use Case

Actions and Replies
A1: News Collector Collect news by searching keyword.
A2: News Collector save news into files.
Level 1.2: Summarize News
Secondary actor: Summarizer, News Presenter.

Goal in context: The diagram refers to the details of the news summarization module of level 1.

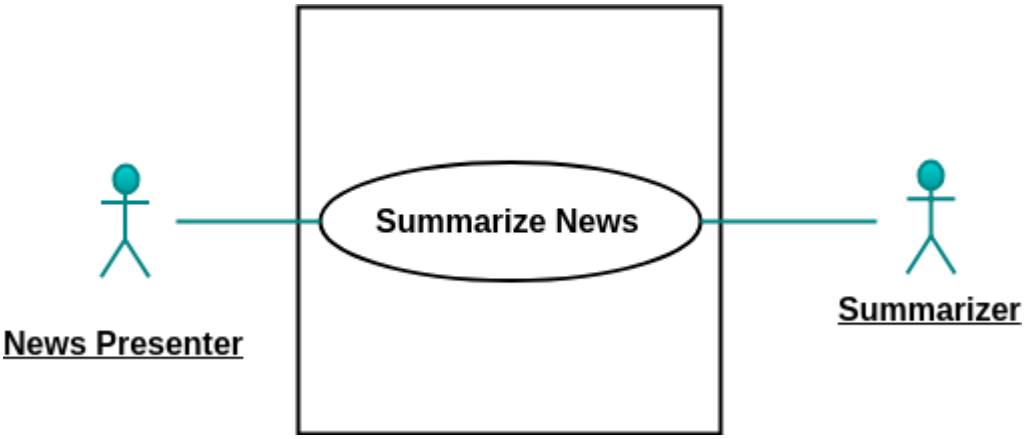

Figure 4: Level 1.2 of Use Case

Actions and Replies
A1: News Presenter provides a news article to summarizer.
R2: Summarizer will return summarized form of the news article as output.

Level 1.3: Show News
Secondary actor: Summarizer, News Presenter
Goal in context: The diagram refers to the details of the show news module of level 1.

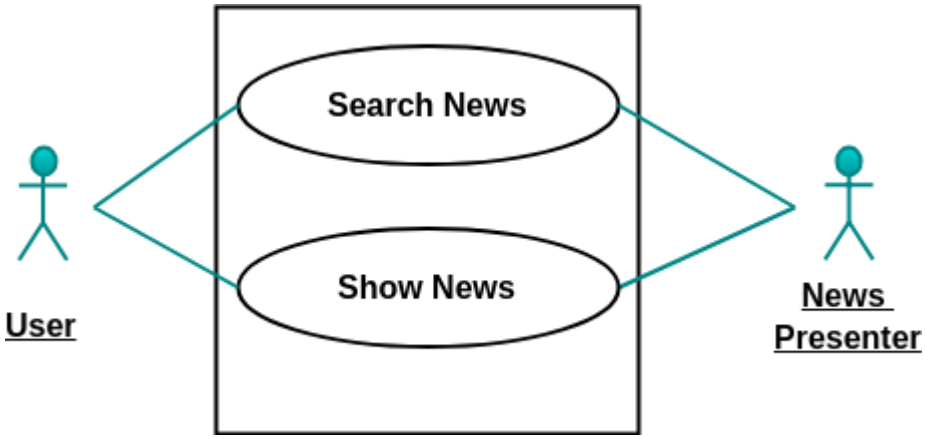

Figure 5: Level 1.3 of Use Case

Actions and Replies
A1: User provides a search keyword.
R1: News Presenter gives show news articles as output.

3.2. Swimlane Diagrams

Level 1.1: Collect News

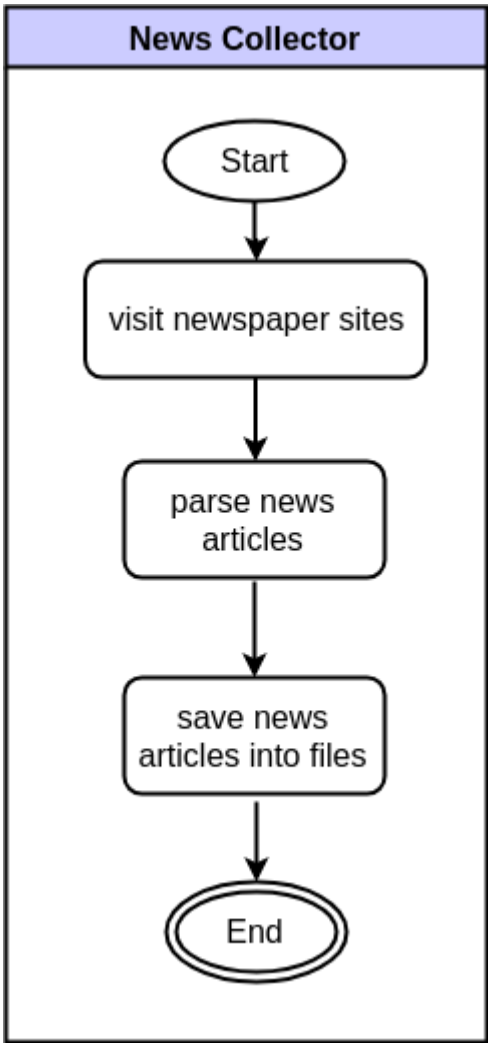

Figure 6: Level 1.1 of Swimlane Diagram

Level 1.2: Summarize News

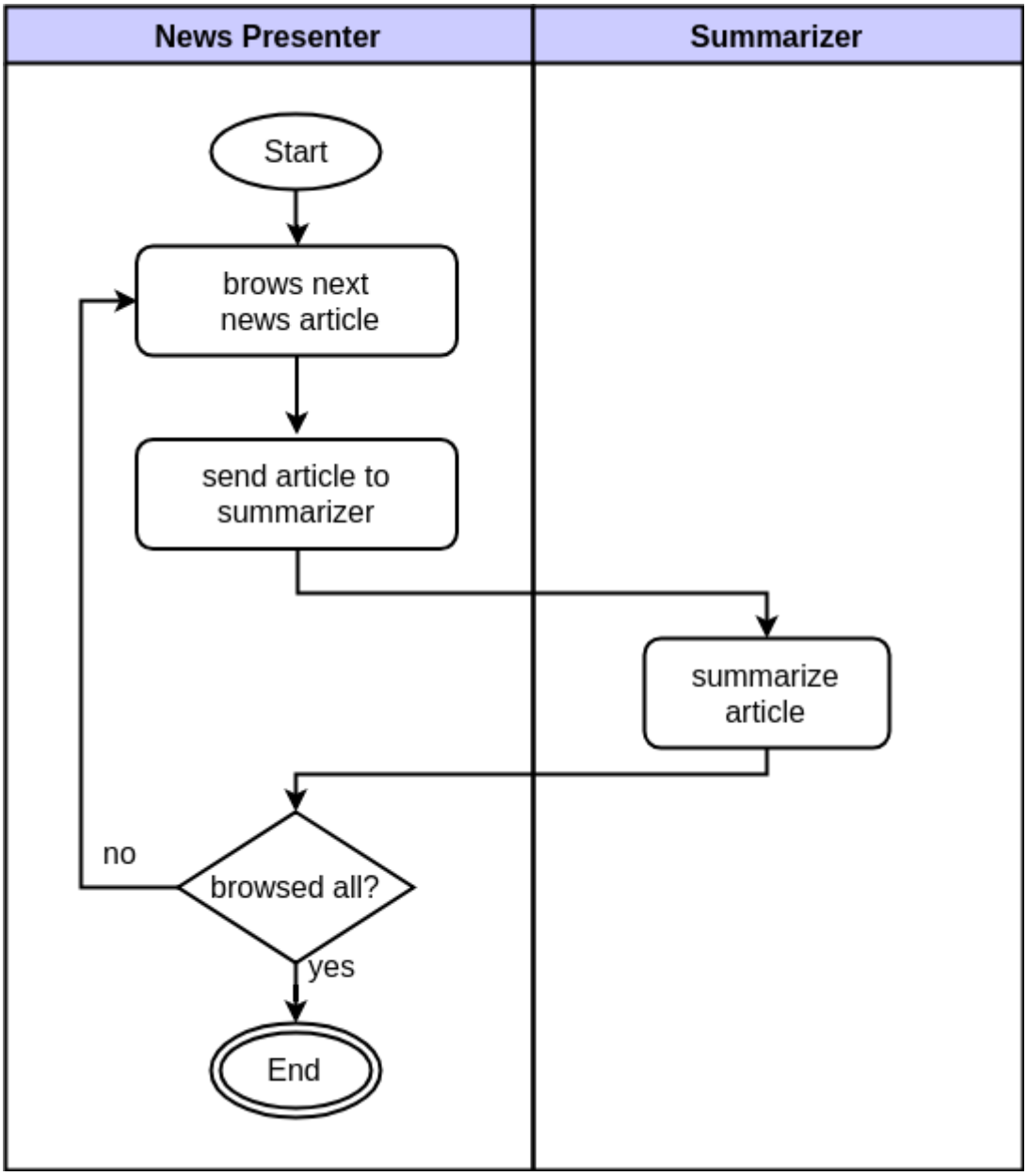

Figure 7: Level 1.2 of Swimlane Diagram

Level 1.3: Show News

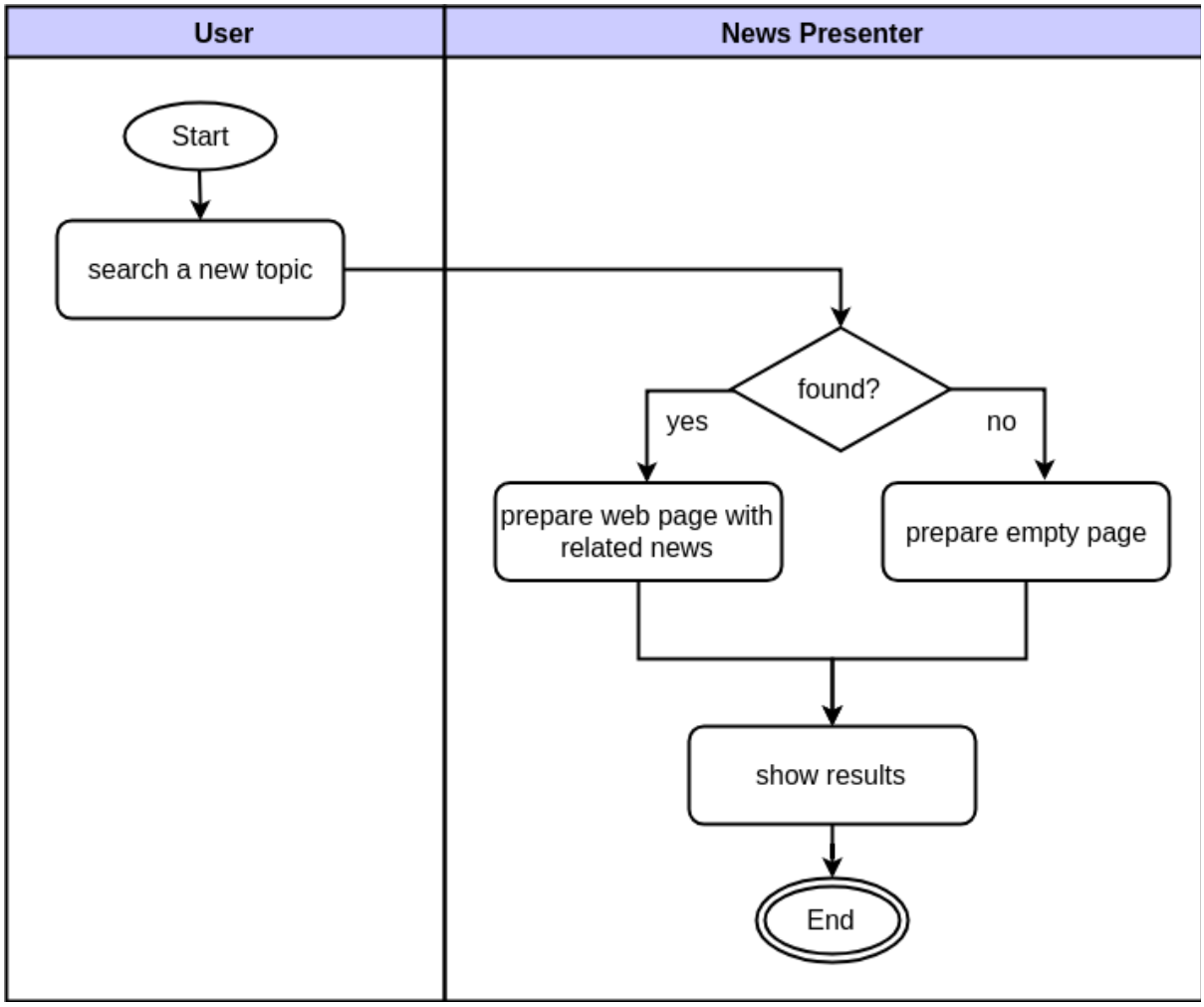

Figure 8: Level 1.3 of Swimlane Diagram

4. Class Based Modeling

4.1. Identifying Analysis Classes

4.1.1. General Classification

Analysis classes manifest themselves in one of the following ways:

- External entities that produce or consume information to be used by a computer-based system.
- Things that are part of the information domain for the problem.
- Occurrences or events that occur within the context of system operation.
- Roles played by people who interact with the system.
- Organizational units that are relevant to an application.
- Places that establish the context of the problem and the overall function of the system.
- Structures that define a class of objects or related classes of objects.

Serial No	**Noun**	**Problem Domain (p)/ Solution Domain (s)**	**General Classification**
1	News Collector	s	role
2	Newspaper	s	thing
3	Article	s	thing
4	Summarizer	s	role
5	News Presenter	s	role

4.1.2. Selection Criteria

Six selection characteristics should be considered for each potential class for inclusion in final class. They are:

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations

6. Essential requirements

Serial No	Noun	Selection Criteria	Remarks
1	News Collector	retained information, needed services	accepted
2	Newspaper	retained information, multiple attributes	accepted
3	Article	retained information, multiple attributes	accepted
4	Summarizer	retained information, needed services	accepted
5	News Presenter	retained information, needed services	accepted

4.2. Final Classes

Serial No	Class	Attributes	Methods
1	News Collector	News	registration() + login() + resetPassword()
2	Newspaper	Article	getArticles()
3	Article	Title, date, url, text, newspaper name	
4	Summarizer		provideNews()+getSummarizedNews()
5	News Presenter	News	provideKeyword()+showNews()

4.3. Class Card

News Collector	
Responsibility	**Collaborator**
Collect News	Newspaper, Article

Newspaper	
Responsibility	**Collaborator**
Get new articles	Article

News Presenter	
Responsibility	**Collaborator**
Show news	Summarizer

4.4. Class Responsibility Collaboration Diagram

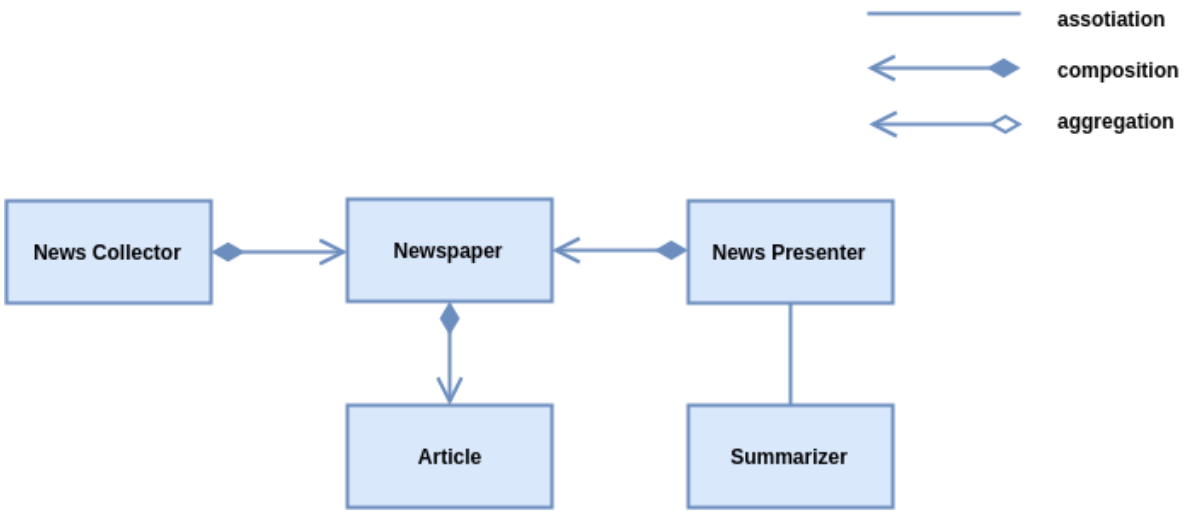

Figure 9: Class Responsibility Collaborator Diagram

5. Architectural Design

As architectural design begins, the software to be developed must be put into context—that is, the design should define the external entities (other systems, devices, people) that the software interacts with and the nature of the interaction. This information can generally be acquired from the requirements model and all other information gathered during requirements engineering. Once context is modeled and all external software interfaces have been described, you can identify a set of architectural archetypes.

5.1. Representing the System in Context

At the architectural design level, a software architect uses an architectural context diagram (ACD) to model the manner in which software interacts with entities external to its boundaries. systems that interoperate with the target system (the system for which an architectural design is to be developed) are represented as

- Superordinate systems
- Subordinate systems
- Peer-level systems

The following diagram represents the software in context

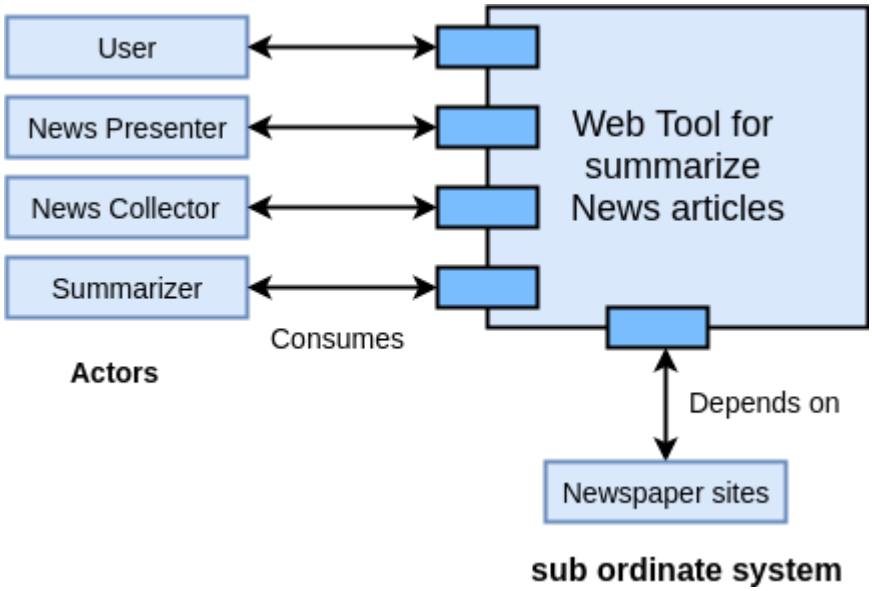

Figure 10: System Architecture in Context

5.2. Refine the Architecture into Components

As the software architecture is refined into components, the structure of the system begins to emerge. The analysis classes introduced in software requirement modeling represent entities within the application domain that must be addressed within the software architecture. Hence, the application domain is one source for the derivation and refinement of components. Another source is the infrastructure domain. The architecture must accommodate many infrastructure components that enable application components but have no business connection to the application domain.

The interfaces depicted in the architecture context diagram imply one or more specialized components that process the data that flows across the interface

For the proposed web tool the following components can be introduced:

- Collecting News
- Showing News Articles
- Summarizing News

Here Summarizing News component is under black box because the procedures and methods for summarization is not final yet.

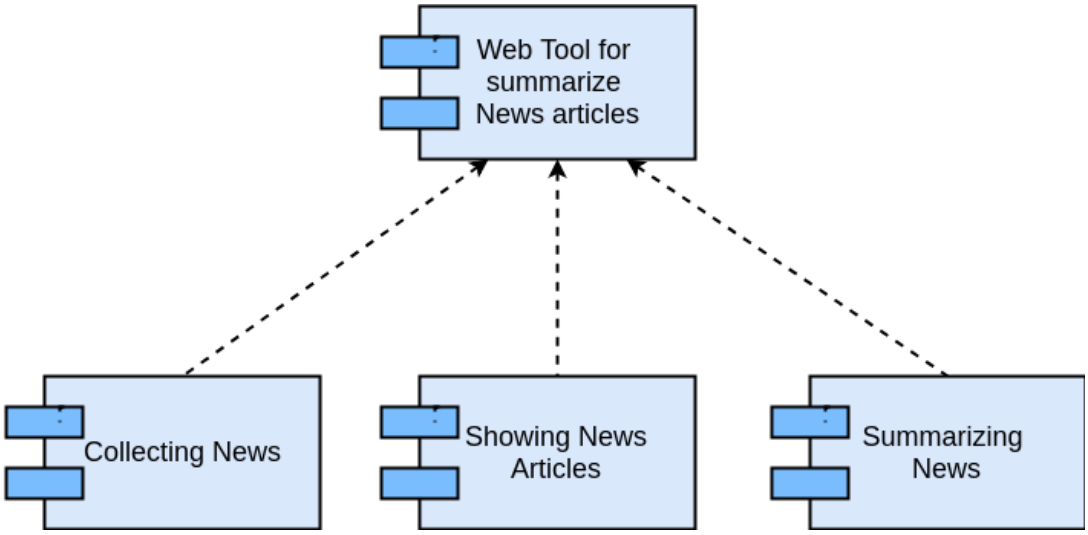

Figure 11: Components of the whole System

6. User Interface Design

User Interface Design is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction.

6.1. Define Interface Objects and Actions

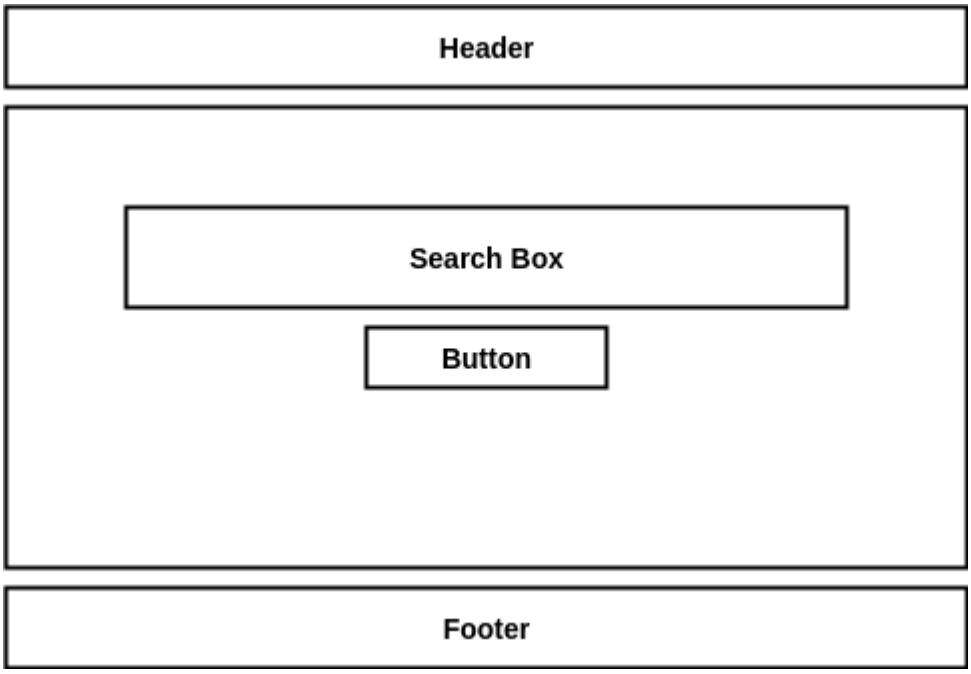

Figure 12: Home Page

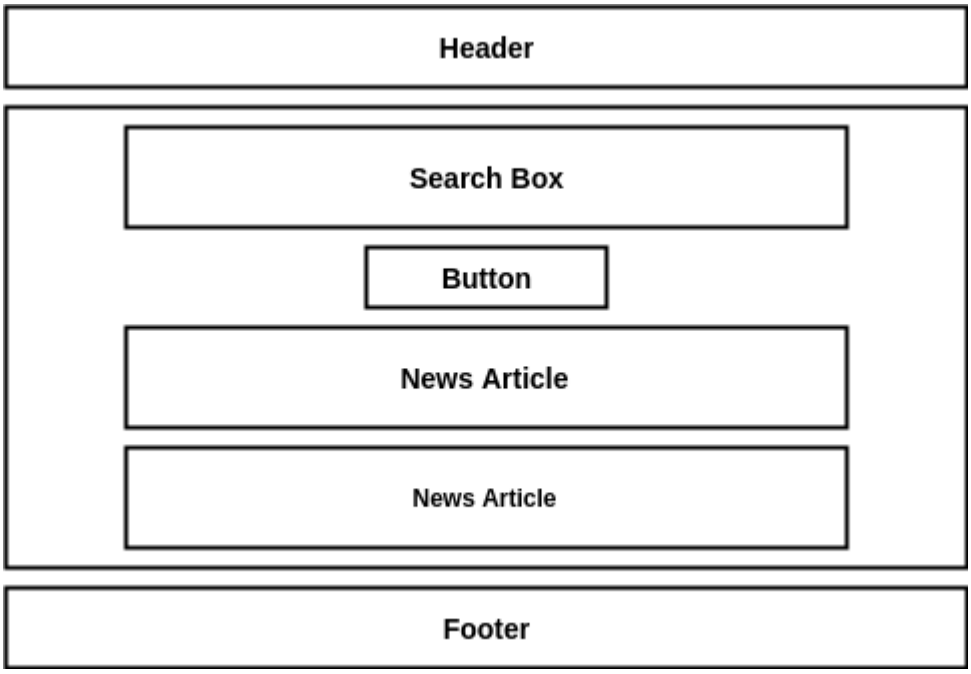

Figure 13: Show News

6.2. Depict each interface state as it look to end user

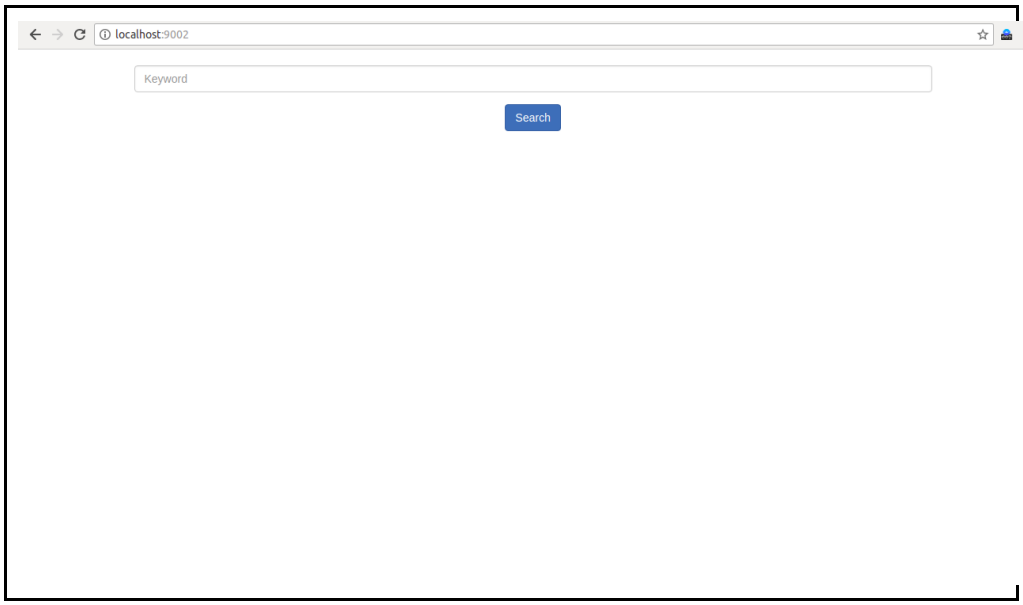

Figure 14: Home Page

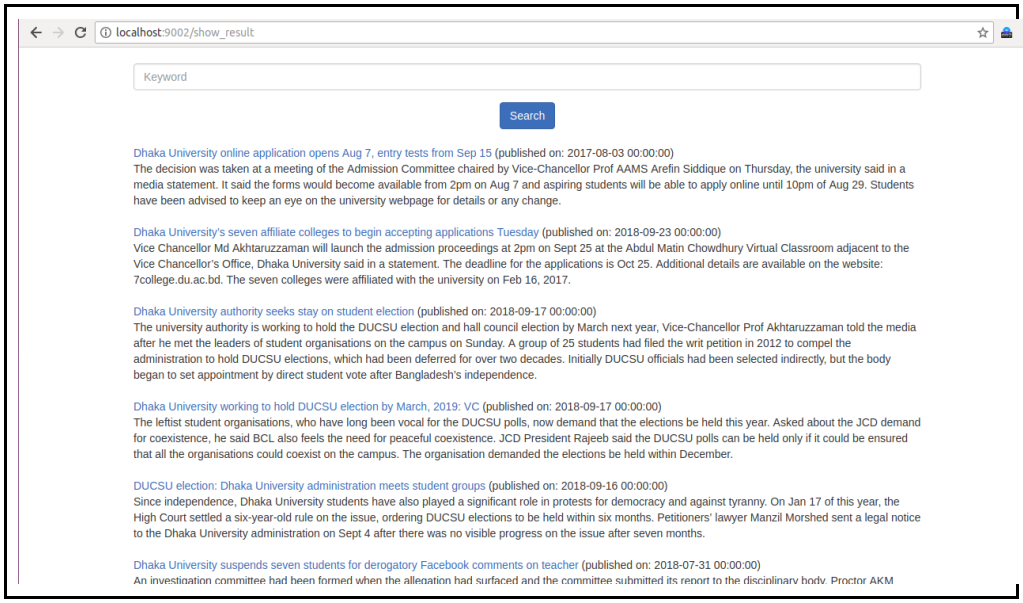

Figure 15: Show News

7. Implementation Overview

The “Web Tool for News Summarization” is an web based tool that can be implemented in server side while any machine having a web browser is able to use the tool via an url. The full server side is implemented with python 3 programming language. On the other hand, client side is implemented only with html, css with some bootstrap dependencies. To build the system I have used some technical terms and features. These terms, features and their constraints are explained in the following.

7.1. Programming Languages

Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossumand first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.
The whole server side source code for this project is implemented with python programming language of version 3.

HTML:

HTML is the standard markup language for creating Web pages. HTML stands for Hyper Text Markup Language. HTML describes the structure of Web pages using markup.
the web pages for client side is built with html.

CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript
The web pages that are built with html have some dependencies with css. those css codes are used for adorning the web pages.

7.2. Frameworks:

Flask:

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if

they were implemented in Flask itself
The server for the web tool is built with Flask framework.

7.3. Tools

jupyter-notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.
All the python source codes are written using jupyter-notebook.

7.4. Libraries

ipynb

ipynb is python library for handling jupyter source codes. it is used in my tool to importing notebook files and using their methods.

flask

python flask library to run the flask server.

textblob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

pickle

It is used for serializing and de-serializing a Python object structure. Any object in python can be pickled so that it can be saved on disk. ... Pickling is a way to convert a python object (list, dict, etc.) into a character stream.

numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical

functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Huguninwith contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

sklearn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

werkzeug

Werkzeug is a comprehensive WSGI web application library. It began as a simple collection of various utilities for WSGI applications and has become one of the most advanced WSGI utility libraries.

8. Source Code Description

The complete source code of the project can be found here - https://github.com/frabbisw/extractive-summarization

This chapter includes short description of our source code which is written in python3, html and css. the server side system is implemented with python and the client side contains html and css.

The main modules of the source codes are followings.

8.1. show news

This is the main module where the routes start. it has 3 methods which are responsible for responses from the routing. The following methods represents them.

search: This method is responsible for the home page. it rendered the home page from templates and return to the client side.
show_result: After user entered a search key this method captures the search key, spit it into words, gathered news finally represents news articles using summarizer module.
show_other_page: because of getting so many news articles based on the searched key the results are splitted into pages and at a time 5 news articles are presented. this method is for presenting news articles at a specific page number. suppose if total searched articles are 100 and page number 5 is requested to be presented, then news articles numbered 21-25 will be shown.

8.2. summarizer

This module has 2 classes named **lexrank_summarizer** and **cluster_based_summarizer**.

8.2.1. lexrank_summarizer

In this class text summarization is implemented using lexrank algorithm introduced by *G Erkan et al*. I implemented these methods the my class for the aproach:

load_idf: To use lexrank approach the IDF(Inverse Document Frequency) of each word is needed. Before the routes start, news_collector module loads idf count of each words from files.
word_to_idf: this method takes a word as input and returns idf of that word as output.
sentence_cosine: this method calculate cosine distance of two sentences. the distance is calculated using sentence cosine formula from that paper I introduced before. "To define similarity, we use the bag-of-words model to represent each sentence as an N-dimensional vector, where N is the number of all possible words in the target language. For each word that occurs in a sentence, the value of the corresponding dimension in the vector representation of the sentence is the number of occurrences of the word in the sentence times the idf of the word. The similarity between two sentences is then defined by the cosine

between two corresponding vectors:

$$\text{idf-modified-cosine}(x, y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x} \text{tf}_{w,y} (\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x} (\text{tf}_{x_i,x} \text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y} (\text{tf}_{y_i,y} \text{idf}_{y_i})^2}}$$

where $tf_{w,x}$ is the number of occurrences of the word w in the sentence S."

sentence_to_matrix: After calculating distances of every pairs of sentences a distance matrix is prepared using the cosine distances.
get_top_rank_sentences: After the matrix is prepared weak bond between sentences are eliminated using a threshold value. Then important sentences are selected as number of neighbour sentences.
summarize: This method takes a passage as input, split the passage into sentences and make a list of important sentences using the method "get_top_rank_sentences". It returns the summarized form of the input passage.

8.2.2. cluster_based_summarizer

I also implement another paper for extractive text summarization named "Extraction based approach for text summarization using k-means clustering" by Ayush et al. In this class the approach is implemented. The following methods are in the class.

summarize: this method is dependent on another class named tfidf_calculator. summarize method takes a passage as input and return summarized form of that passage. firstly it split passage into sentences and calculate tf-idf score of that sentences. Then it creates a cluster based on the tf-idf scores of sentences. The approach is described from their paper
"The major idea is to divide the entire document into sentences. Each sentence can be considered as a point in the Cartesian plane.
Each sentence is then broken into tokens and the tf-idf score is computed for each token in the sentence.
Tft = f(t,d) / f(d) where , t is a token
d represents the document f(t,d) represents frequency of t in d f(d) represents frequency of every term in d
idft = log10 (N / f(t,d))
where , N is the number of sentences in the document
tf-idf t = Tf t * idf
t Score for each sentence is computed by summing up the tf-idf score for every token in the sentence and normalizing it by using the sentence length.
Score(X) = ∑t tf-idft / |X| where , X represents a sentence in the document
t is a term in X
|X| represents length of X"

Besides another helper class "**tfidf_calculator**" is needed for the cluster_based_summarization. The methods of the class are followings:

load_idf: To use lexrank approach the IDF(Inverse Document Frequency) of each word is needed. Before the routes start, news_collector module loads idf count of each words from files.
word_to_tf: takes a word as input and return tf of the word.
word_to_idf: takes a word as input and return idf of the word.
word_to_tf_idf: takes a word as input and return tf*idf of the word.
sentence_tfidf: takes a sentence as input and return tf-idf of that sentence.
get_sent_tfidfs: takes a list of sentences and return list of tf*idf of those sentences.

8.3. news_collector

Before all of the routes start this news_collector class collect news from some of newspaper sites. this class store news to files following these steps:

- visits newspaper sites and captures all of news using **newspaper3k** library.
- parses news articles and makes list of articles.
- dump list of articles to local storage as python pkl file.

8.4. store_idf

These steps are followed to store idf of the words:

- load news articles from local storage.
- split articles into words.
- make each word into lower word.
- calculate idf of each word using this formula-

$$\mathrm{idf}_i = \log\left(\frac{N}{n_i}\right)$$

where N is the total number of the documents in a collection, and ni is the number of documents in which word i occurs.

8.5. templates

This module consists of some web pages that are represented to clients. The web pages are followings:

search.html: A html page that has a search bar and a submit form. after submitting the form user get the show_result route.
show_news.html: A html page that contains the news articles with information.

8.6. newspaper classes

There are 3 classes that are created to retain information about the news.

8.6.1. paragraph

This class is for retaining information about a article. it contains articles attributes called url, title, text, published date, newspaper brand.

8.6.2. customizedPaper

This class is for retaining information about a newspaper. It contains all articles of newspaper.

8.6.3. idfBlob

This class retains informatin of idf counts of each word. other modules get idf of word using a method called **getIdf** in this class.

8.7. prepare single doc news list

Following methods of this module to prepare summarized article list.

- **load_news:** load nws articles that saved by news collector module.
- **getSummarizedFromList:** Takes a list of articles as input and return list of summarized news articles.
- **getNewsFromKey:** Takes a search key as input, split key into words and search news with this words and finally return list of news articles.

8.8. prepare multi doc summarized news

Following methods of this module to prepare summarized article list.

- **load_news:** load nws articles that saved by news collector module.
- **getSummarizedFromList:** Takes a list of articles as input, create a blob of articles and finally return summarized version of that blob of news articles.
- **getNewsFromKey:** Takes a search key as input, split key into words and search news with this words and finally return list of news articles.

9. Test Plans

This chapter presents test plan of the project

Test Plan Identifier:

Web Tool for News Summarization Test Plan 1.

Introduction:

This test plan has been developed for "Web Tool for News Summarization". The whole system is divided into only one portion: search news articles by keywords.

Test-Item to be tested:

I tested the Web Tool for News Summarization. Software Requirement Analysis and Specification Document of the system will be used for this purpose.

Features to be tested:

The following features are tested:

- Search Bar from Homepage
- Search Bar from show results page
- Pager items in show results page

Features not to be tested:

All the features of the system are tested.

Approach:

I used end to end automated (integration) testing technique to test the system.

Item Pass/Fail Criteria:

If actual output of a test case does not match with expected output of the test case, the test case is considered as failed. 100% of all test cases should pass. No failed case should be crucial to the end-user's ability to use the application.

Test Deliverables:

I will deliver test plan document, test case and test report.

Scheduling:

Scheduling is given below with different part of Quality Assurance (QA) and duration -

QA	Duration
Test Plan	2 days
Testing	2 days

Planning Risks and Contingencies:

The following scenarios are considered as risks for the project:

- Delay in requirements engineering.
- Delay in developing.
- Modification in development technology.

Test Cases

Search Bar of Home page

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
HT01	Check Search Bar from Home page with valid data	go to url enter keyword select mode click search button	keyword = bangladesh west indies mode= single news summarization	Summarized News should be arrived based on the topic.	As Expected	Pass
HT02	Check Search Bar from Home page with valid data	go to url enter keyword select mode click search button	keyword = bangladesh west indies mode= multiple news summarization	Summarized News should be arrived based on the topic.	As Expected	Pass
HT03	Check	go to url	keyword =	No result	As	Pass

	Search Bar from Home page with empty data	enter keyword select mode click search button	null mode= multiple news summarization		Expected	

Search Bar of Show Result page

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
ST01	Check Search Bar from Home page with valid data	go to url enter keyword select mode click search button	keyword = bangladesh west indies mode= single news summarization	Summarized News should be arrived based on the topic.	As Expected	Pass
ST02	Check Search Bar from Home page with valid data	go to url enter keyword select mode click search button	keyword = bangladesh west indies mode= multiple news summarization	Summarized News should be arrived based on the topic.	As Expected	Pass
ST03	Check	go to url	keyword =	No result	As	Pass

	Search Bar from Home page with empty data	enter keyword select mode click search button	null mode= multiple news summarization		Expected	

10. User manual

The web tool is developed using users' requirements. According to my software requirement specifications and analysis, I tried my best to fulfill all these requirements.

In order to use the tool, a non-technical user needs to learn the manual. The web tool can be run with any device having a web browser.

10.1. Entering web url

To use the tool at first a user needs to open a web browser and enter the provided url (The url with port number will be given to user). Here is a snapshot of entering the url from localhost.

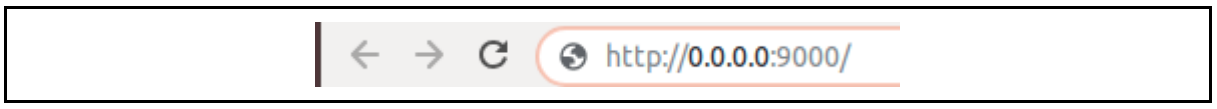

Figure 16: Entering web url

10.2. Home Page

Entering the url an user will find himself with the home page. the home page comprises 1 input field for keyword, 1 select item for summarization mode and a submit button named **search**. by clicking the submit button user will find the search results.

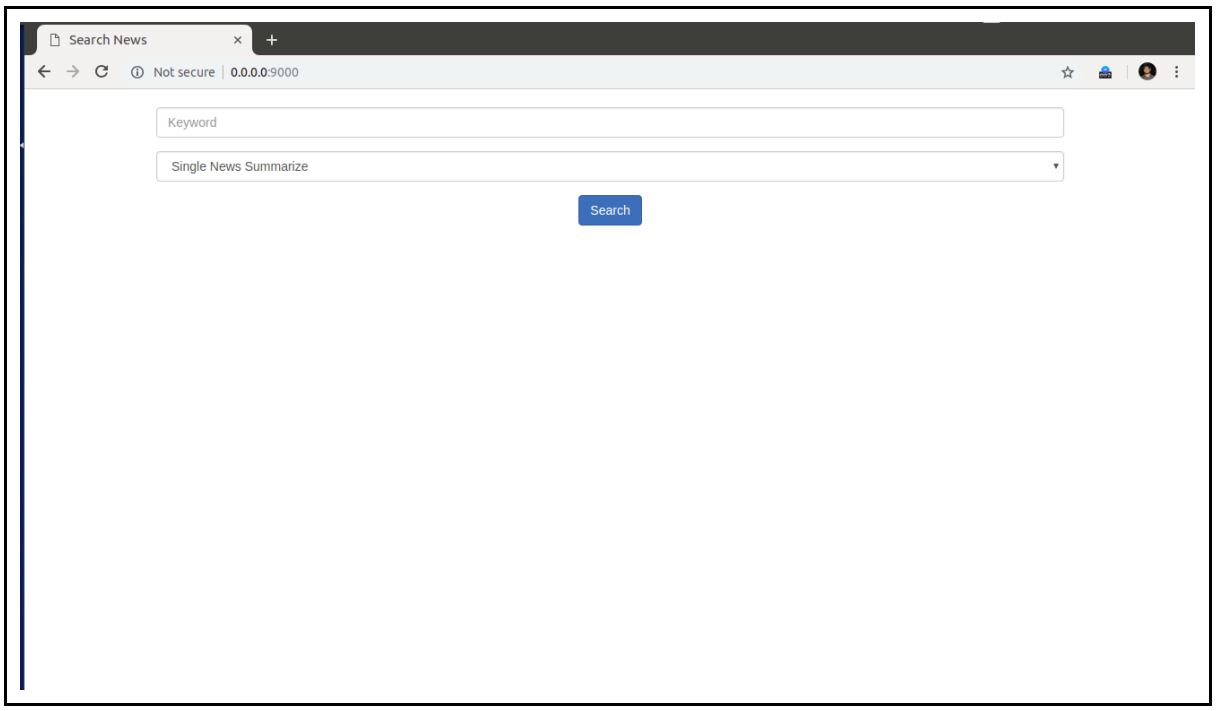

Figure 17: Home Page

10.3. Entering Keyword

User has to enter the input field of keyword. user can enter one or multiple keywords. the news articles which titles matched more with keywords will arrive earlier in the search result.

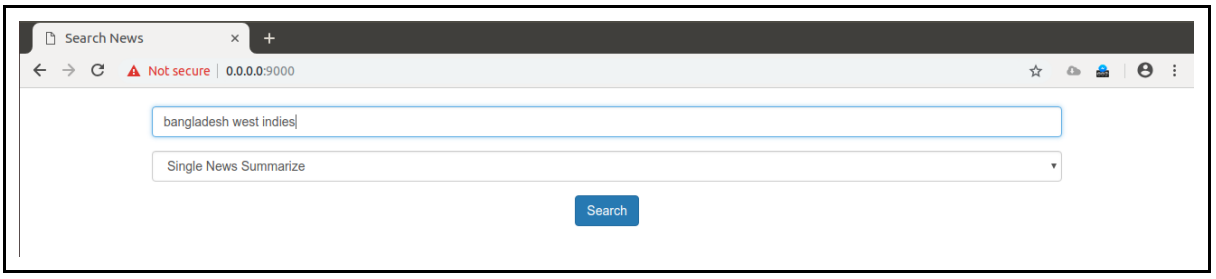

Figure 18: Entering Keyword

10.4. Selecting Summarization mode

After entering keyword user needs to select summarization mode. There are two modes here: Single News Summarization and Multiple News Summarization mode. Single News Summarization will provide individual summarized form of news articles where Multiple News Summarization will provide combined summary of searched articles.

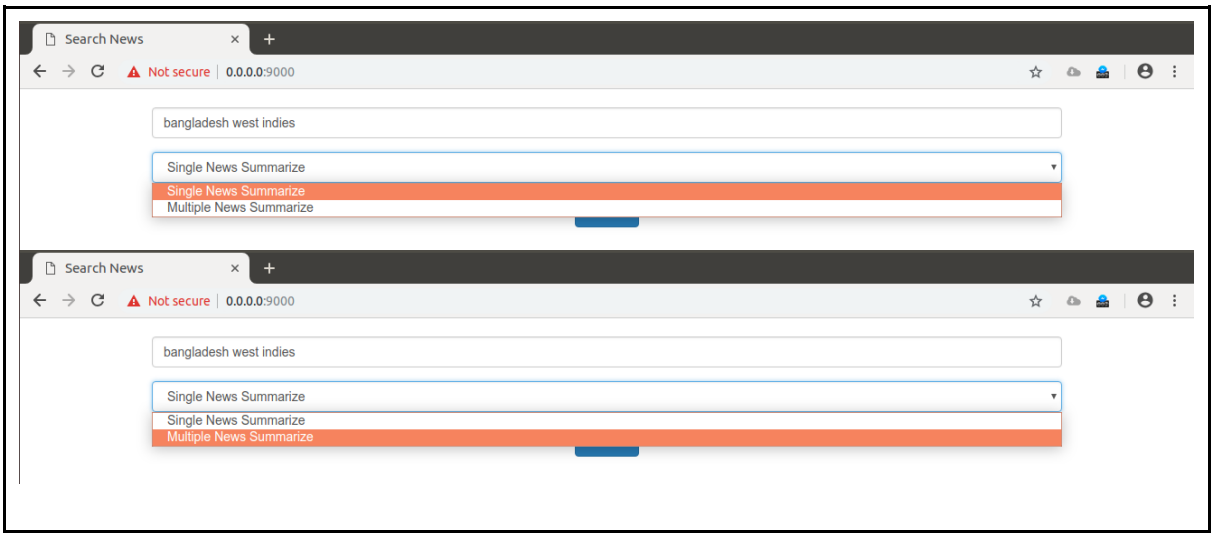

Figure 19: Selecting Summarization mode

10.5. Clicking Search button

When the fields are filled up user is one step away from the result. The user needs to click the search button to get the next page.

Figure 20: Clicking Search Button

10.6. Search Result

User can view two types of results based on his selecting mode. Besides the search bar of the home page is also presented here advantages of the user. The following types of searched results will be presented to user.

10.6.1. Single News Summarization

Searched page

The searched page will be looked like this for Single News Summarization.

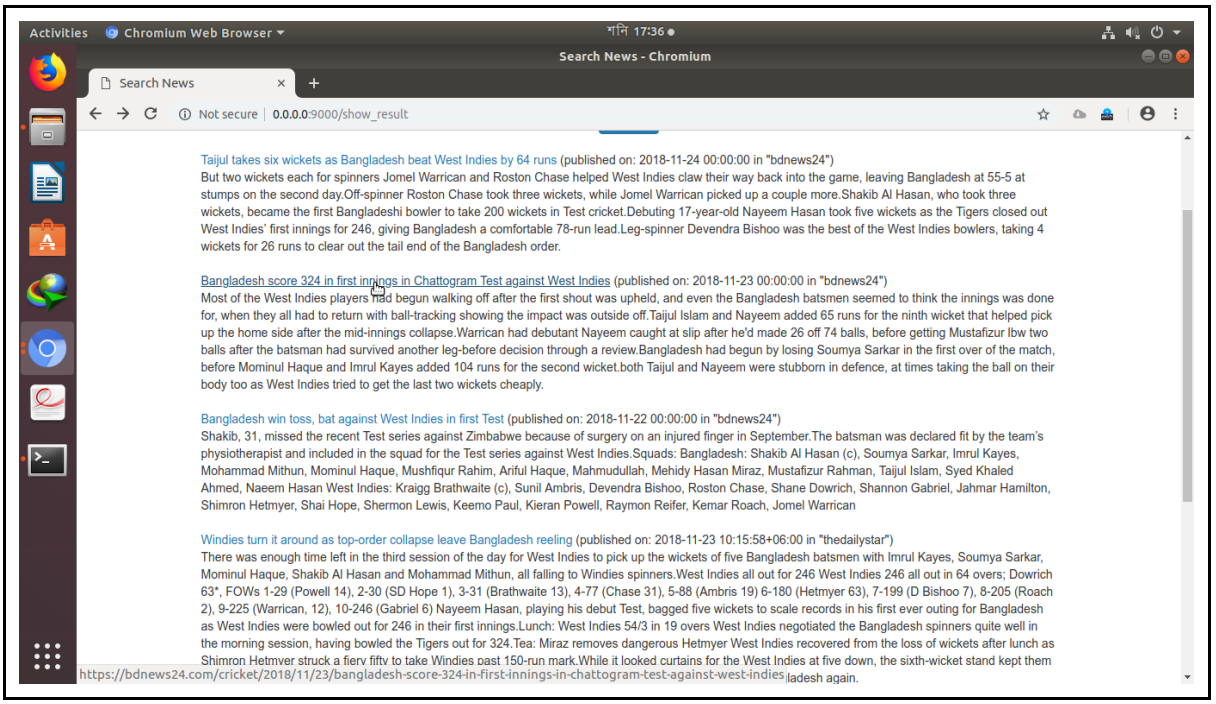

Figure 21: Searched Page of Single News Summarization

Content Information

For Single News Summarization, The paragraphs of the page have following informations: News title, published date, newspaper name and summarized text.

Bangladesh score 324 in first innings in Chattogram Test against West Indies (published on: 2018-11-23 00:00:00 in "bdnews24")
Most of the West Indies players had begun walking off after the first shout was upheld, and even the Bangladesh batsmen seemed to think the innings was done for, when they all had to return with ball-tracking showing the impact was outside off.Taijul Islam and Nayeem added 65 runs for the ninth wicket that helped pick up the home side after the mid-innings collapse.Warrican had debutant Nayeem caught at slip after he'd made 26 off 74 balls, before getting Mustafizur lbw two balls after the batsman had survived another leg-before decision through a review.Bangladesh had begun by losing Soumya Sarkar in the first over of the match, before Mominul Haque and Imrul Kayes added 104 runs for the second wicket.both Taijul and Nayeem were stubborn in defence, at times taking the ball on their body too as West Indies tried to get the last two wickets cheaply.

Bangladesh score 324 in first innings in Chattogram Test against West Indies (published on: 2018-11-23 00:00:00 in "bdnews24")
Most of the West Indies players had begun walking off after the first shout was upheld, and even the Bangladesh batsmen seemed to think the innings was done for, when they all had to return with ball-tracking showing the impact was outside off.Taijul Islam and Nayeem added 65 runs for the ninth wicket that helped pick up the home side after the mid-innings collapse.Warrican had debutant Nayeem caught at slip after he'd made 26 off 74 balls, before getting Mustafizur lbw two balls after the batsman had survived another leg-before decision through a review.Bangladesh had begun by losing Soumya Sarkar in the first over of the match, before Mominul Haque and Imrul Kayes added 104 runs for the second wicket.both Taijul and Nayeem were stubborn in defence, at times taking the ball on their body too as West Indies tried to get the last two wickets cheaply.

Bangladesh score 324 in first innings in Chattogram Test against West Indies (published on: 2018-11-23 00:00:00 in "bdnews24")
Most of the West Indies players had begun walking off after the first shout was upheld, and even the Bangladesh batsmen seemed to think the innings was done for, when they all had to return with ball-tracking showing the impact was outside off.Taijul Islam and Nayeem added 65 runs for the ninth wicket that helped pick up the home side after the mid-innings collapse.Warrican had debutant Nayeem caught at slip after he'd made 26 off 74 balls, before getting Mustafizur lbw two balls after the batsman had survived another leg-before decision through a review.Bangladesh had begun by losing Soumya Sarkar in the first over of the match, before Mominul Haque and Imrul Kayes added 104 runs for the second wicket.both Taijul and Nayeem were stubborn in defence, at times taking the ball on their body too as West Indies tried to get the last two wickets cheaply.

Bangladesh score 324 in first innings in Chattogram Test against West Indies (published on: 2018-11-23 00:00:00 in "bdnews24")
Most of the West Indies players had begun walking off after the first shout was upheld, and even the Bangladesh batsmen seemed to think the innings was done for, when they all had to return with ball-tracking showing the impact was outside off.Taijul Islam and Nayeem added 65 runs for the ninth wicket that helped pick up the home side after the mid-innings collapse.Warrican had debutant Nayeem caught at slip after he'd made 26 off 74 balls, before getting Mustafizur lbw two balls after the batsman had survived another leg-before decision through a review.Bangladesh had begun by losing Soumya Sarkar in the first over of the match, before Mominul Haque and Imrul Kayes added 104 runs for the second wicket.both Taijul and Nayeem were stubborn in defence, at times taking the ball on their body too as West Indies tried to get the last two wickets cheaply.

(Title, published date, newspaper name and summarized text)

Figure 22: Contents of Searched Result for Single News Summarization

clicking on news title

By clicking on any news title user will find the source article.

Taijul takes six wickets as Bangladesh beat West Indies by 64 runs (published on: 2018-11-24 00:00:00 in "bdnews24")
But two wickets each for spinners Jomel Warrican and Roston Chase helped West Indies claw their way back into the game, leaving Bangladesh at 55-5 at stumps on the second day.Off-spinner Roston Chase took three wickets, while Jomel Warrican picked up a couple more.Shakib Al Hasan, who took three wickets, became the first Bangladeshi bowler to take 200 wickets in Test cricket.Debuting 17-year-old Nayeem Hasan took five wickets as the Tigers closed out West Indies' first innings for 246, giving Bangladesh a comfortable 78-run lead.Leg-spinner Devendra Bishoo was the best of the West Indies bowlers, taking 4 wickets for 26 runs to clear out the tail end of the Bangladesh order.

Figure 23: Clicking on News Title

show other pages

User can visit other articles by clicking those buttons.

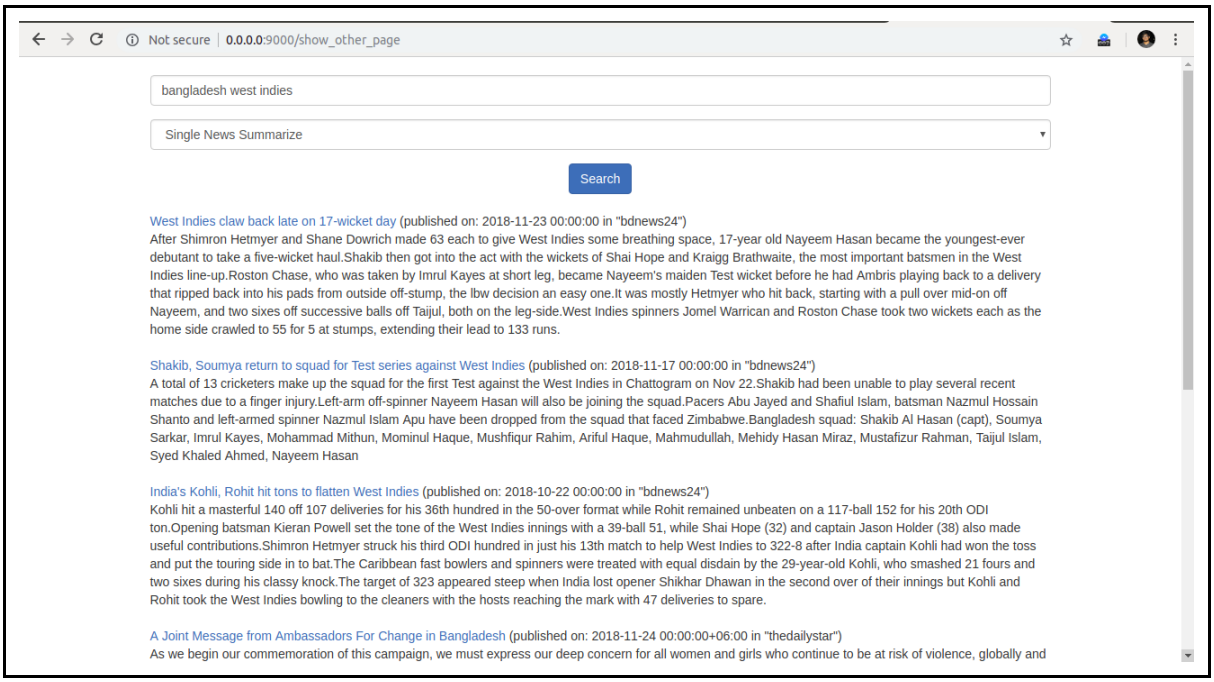

Figure 24: Show other page by clicking a page number

10.6.2. Multiple News Summarization

User will find a page like this if he selects multiple news summarization option. The page has combined summarization text of searched news articles.

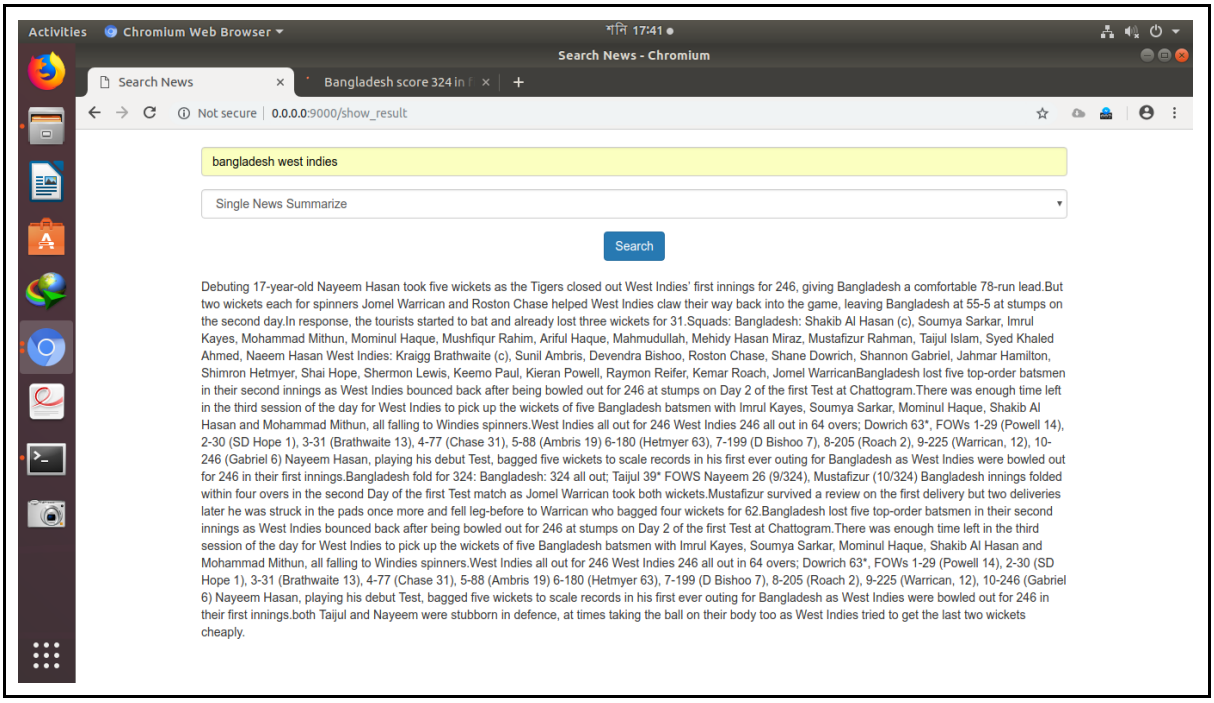

Figure 25: Content of Multiple News Summarization

11. Conclusion

It was so much challenging to prepare a final report for the first time. I think that this report has been written in an easy-to-read way as well as with full information required to have a good concept over the idea. The reader of should easily understand the information of the report.

12. Reference

1. Software Engineering: A Practitioner's Approach 7th Edition by Roger S. Pressman.
2. "CSS developer guide". Mozilla Developer Network. Retrieved 2015-09-24.
3. Flanagan, David. JavaScript - The definitive guide (6 ed.). p. 1. JavaScript is part of the triad of technologies that all Web developers must learn: HTML to specify the content of web pages, CSS to specify the presentation of web pages, and JavaScript to specify the behaviour of web pages.
4. https://web.archive.org/web/20171117015927/http://flask.pocoo.org:80/docs/0.10/foreword
5. http://jupyter.org/
6. https://scikit-learn.org/
7. https://pypi.org/project/Werkzeug/
8. Erkan, G. and Radev, D.R., 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, pp.457-479.
9. Agrawal, A. and Gupta, U., 2014. Extraction based approach for text summarization using k-means clustering. *Int. J. Sci. Res. Publ.(IJSRP)*, *4*(11).