Partial Project Report

**Document Summarization and Mind Map**

Submitted to the

Savitribai Phule Pune University, Pune

In partial fulfilment for the award of the Degree of

Bachelor of Engineering in Information Technology

By

|  |  |
| --- | --- |
| 407002 | Aishwarya Kore |
| 407003 | Akriti Singh |
| 407024 | Chirag Rajpal |
| 407042 | Abhijeet Ingale |

Under the guidance of

Dr. A. N. Bhute



Department of Information Technology

Sinhgad College of Engineering

S. No.44/1, Off. Sinhgad Road, Vadgaon (Bk.), Pune-411041, Maharashtra, India

(2017 - 2018)

**Acknowledgement**

WE are highly indebted to our guide Dr. A. N. Bhute for his guidance and constant supervision as well as for providing necessary information regarding the report and also for their support in completing the report.

This acknowledgement would be incomplete without expressing our special thanks to Prof. G. R. Pathak, Head of the Department (Information Technology) for his support during the work.

We would like to extend our heartfelt gratitude to our Principal, Dr. S. D. Lokhande who provided a lot of valuable support, mostly being behind the veils of college bureaucracy.

We would also like to express our gratitude towards our parents and friends for their kind co-operation and encouragement which help us in completion of this report. Our thanks and appreciations also go to our colleague in developing the report and people who have willingly helped me out with their abilities.

**INDEX(generate it automated one)**

1. **INTRODUCTION 5**
   1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   2. Aim . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   3. Objective . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   4. Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
2. **LITERATURE SURVEY 7**
3. **REQUIREMENT SPECIFICATION 9**
4. **ANALYSIS AND DESIGN 10**
   1. Architecture of Document Summarization. . . . . . . . . . . . . . . . . . 10
   2. Analysis and Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
      1. USE CASE DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . 11
      2. CLASS DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
      3. SEQUENCE DIAGRAM . . . . . . . . . . . . . . . . . . . . . . . . . 14
      4. STATE DIAGRAM. . . . . . . . . . . . . . . . . . . . . . . . . 15
      5. ACTIVITY DIAGRAM. . . . . . . . . . . . . . . . . . . . . . . . . . . 16
5. **APPLICATION 17**
6. **CONCLUSION 18**

**REFERNCES 19**

**List of Figures**

* 1. Use Case Diagrams . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
     + 1. Use Case Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
       2. Class Diagram. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
       3. Sequence Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
       4. State Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
       5. Activity Diagram. . . . . . . . . . . . . . . . . . . . . . . 1

**ABSTRACT**

In this era of internet and huge data available on it, we need to understand all information in a faster and efficient way so that we don’t miss important information. This need of understanding all the information creates the requirement for automation. Text Summarization helps to achieve our goal. It generates summary of the text (which consists of important information) having meaning and no redundancy. The aim of this project is to study different algorithms for text summarization, to analyze the Text summarization, to design and Implement Text summarization using abstractive or extractive based techniques, to generate concise summaries of complex articles and documents and to generate the graphical representation of the text which is called as the mind map of the text document. Mind Map helps the reader to get a better and faster understanding of the text document.

**CHAPTER 1**

**INTRODUCTION**

This system generates summary of a text document for better understanding and further generates the graphical Mind Map. The summary consists of the important sentences within the document. These sentences when combined together generate complete meaning of the document but in less number of words. The summary of the document is generated by extracting some of the features of the text like sentence length, position, frequency etc. Sentences are also processed for finding out the cosine between them and thus to avoid redundancy. Humans have a tendency to understand graphical contents easier than in text format. After the generation of summarized document, the system generates a graphical Mind Map of the summary. Mind Map consists of cluster of nodes representing the keywords which are mapped in way that gives the end user a complete understanding of the document.

* 1. **AIM AND OBJECTIVES**

**Aim:**

To design and develop an automated system which generates document summarization and Mind Map.

**Objective:**

* To study different algorithms for text summarization.
* To analyse the Text summarization.
* To design and Implement Text summarization using abstractive or extractive based techniques, to generate concise summaries of complex articles and documents.
* To generate the Mind Map of a large document to help the user understand the document in a better and faster way.

**CHAPTER 2**

**LITERATURE SURVEY**

1. The paper [1], Moratanch, N., et.al. explained the Extractive Summarization extracts important sentences or phrases from the source documents and group them to generate summary without changing the source text. It also selects informative sentences from the document as they exactly appear in source based on specific criteria to form summary. It basically assigns a score to each sentence based on feature then rank sentences according to their score. Sentences with the highest score are likely to be included in final summary. Term frequency (TF) and the inverse document frequency (IDF) are numerical statistics presents how important a word in a given document. TF is number of times a term occurs in the document and IDF is a measure that diminishes the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely.
2. The paper [2], Abdeen, M., et al. explained the important and ways to generate a Mind Map. A mind map represents ideas, words, text, images around a central idea. It captures these ideas quicker than expressing them using only words or phrases. It is basically a Graphical summary of a large document. He also explained the modules of the tool used i.e., The M^2Gen tool. It consists of five modules:

• Morphological Analyzer

• Parser

• Syntax Analyzer

• Semantic Analyzer (which further consists of three sub-modules)

• Mind map conversion

1. The Paper [3], Mohit Agarwal, et al. proposed an artificial neural network and genetic algorithm to solve text recognition problem. A hetero-associative neural networkis proposed that will train the system for deciphering from jpeg images into digitsso that characters are recognized easily. The genetic algorithm repeatedly performs crossover on sections and parts of text data from an image file to train the system using trained patterns available. According to author NEURAL NETWORKSor Artificial neural networks (ANNs) are a family of statistical learning models. ANNs are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Hetero-associative neural network has been used here because they can recall an associated pieceof datum from one category upon presentation of data from another category, helping in image classification. Hetero-associative neural network to convert the image to digits.
2. The paper [4],Krishnan, Praveen, et al. presented a deep feature representation for handwritten text recognition and retrieval. They used synthetically generated large handwritten data sets and enabled large scale learning over it because data is not limited. The model embedded both the CNN (Convolution Neural Networks) representation and text labels into a common subspace where both the images and their corresponding text representation lie close to each other. In CNN architecture, we capture local substructure of an image by constraining each neuron to depend only on a spatially local subset of the variables in the previous layer. For example, if the input to the CNN is a 32-by-32 image patch, a neuron in the first hidden layer might only depend on an 8-by-8 sub-window within the overall 32-by-32 window. Thus, in a CNN, individual neurons generally have a local receptive field rather than a global receptive field. In terms of network architecture, this translates to a sparser set of edges since adjacent layers are not always fully connected.
3. The paper [5], Rabi, Mouhcine, et al. explained Hidden Markov models (HMMs). They are usually used for classification. They avoid the the need for explicit segmentation. HMMs have sound mathematical and theoretical foundations. Each word is described by a model built by concatenating the models of the component character. HMM is a doubly stochastic process with an under-lined stochastic process (Markov chain) that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observed symbols. HMM is defined as ,

λ = (N, M, A, B, П) :

• N: The number of states of the model,

• M: The number of observation symbols in the alphabet,

• A: A set of state transition probabilities,

• B: A probability distribution in each of the states,

• П: The initial state distribution

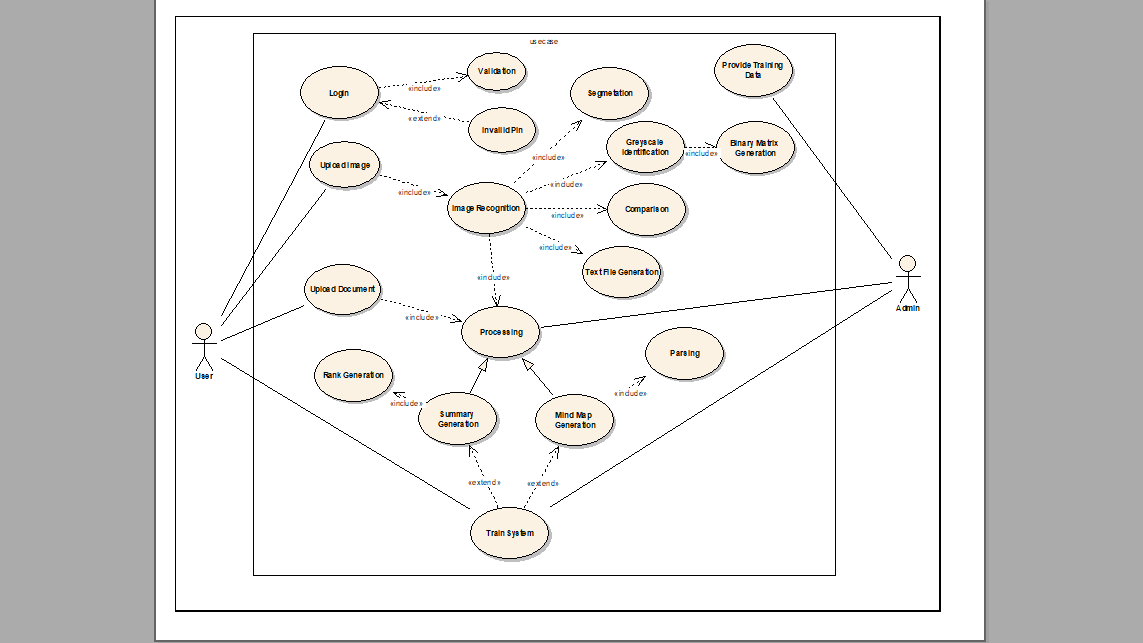
**CHAPTER 3**

**SPECIFICATIONS**

* 1. **HARDWARE REQUIREMENTS**
     1. Processor : Intel Pentium or higher
     2. RAM : 500 MB or higher
     3. Monitor: Keyboard and Mouse
     4. Hard Disk: 20 GB or higher
  2. **SOFTWARE REQUIREMENTS**
     1. OS : Windows 7 or higher
     2. Java Installed on machine with jdk1.8 or higher
     3. Python 3.2 installed on machine or higher

**CHAPTER 4**

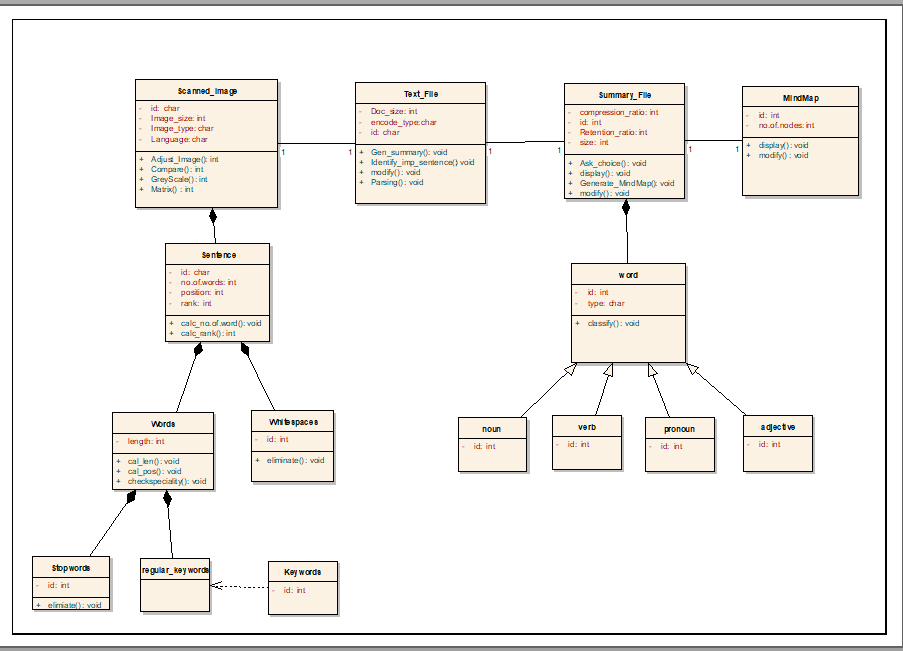
**SYSTEM ANALYSIS AND DESIGN**

**4.1.1 Use Case Model:** A use case diagram at its simplest is a representation of user’s interaction with the system that shows the relationship between the user and different use cases in which the user is involved. A use case diagram can identify the different type of users of a system a system and the different use cases and will often be accompanied by other types of diagram as well. [](use%20case.png)

**USE CASE DOCUMENTATION**

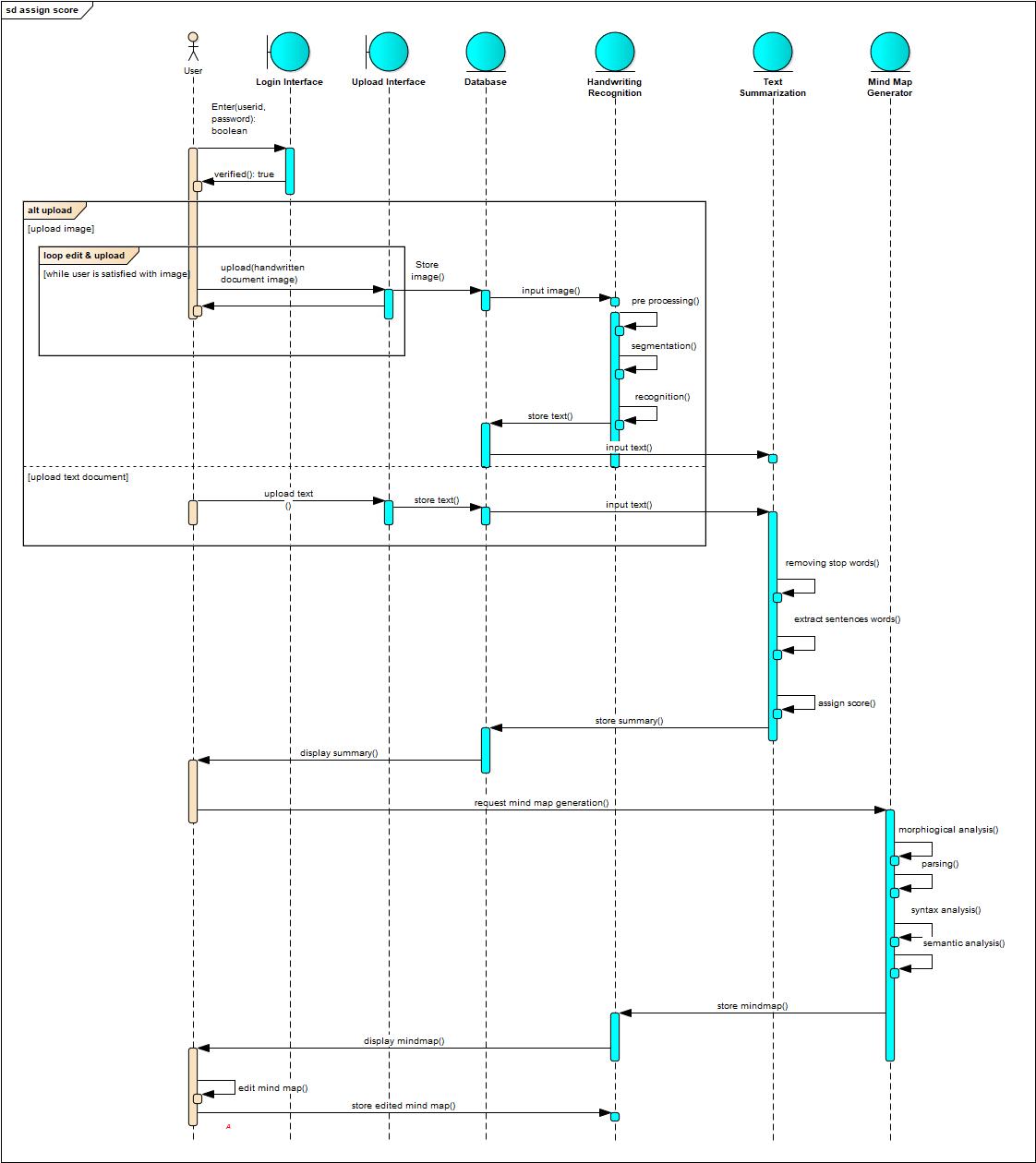
|  |  |  |  |
| --- | --- | --- | --- |
| **SR. NO.** | **USE CASE ID** | **USE CASE NAME** | **DESCRIPTION** |
| 1 | Login | LOGIN | For authentic user login |
| 2 | Val | Validation | For Validation of User Credentials |
| 3 | Invl\_pin | Invalid Pin | For Handling Invalid Acces |
| 4 | Upl\_image | Upload Image | For Uploading Handwritten Text Image |
| 5 | Img\_reg | Image Recognition | For Image Recognition |
| 6 | seg | Segmentation | For Pixel segmentation of Image |
| 7 | Grey\_scale | Grey Scale | For Grey scale Recognition of each pixel |
| 8 | Text\_file | Text File Generation | For generation of text file |
| 9 | Upl\_doc | Upload Document | For Direct uploading text document |
| 10 | Sum\_gen | Summary Generation | For generating summary from text |
| 11 | Mind\_gen | Mind Map Generation | For generating mind map from summarized text |
| 12 | Train\_data | Providing Training Data | For Providing training data for training system |

**4.2.2Class Diagram:** In software engineering, a class diagram is Unified Modeling Language(UML) is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations(or methods), and the relationships among objects.

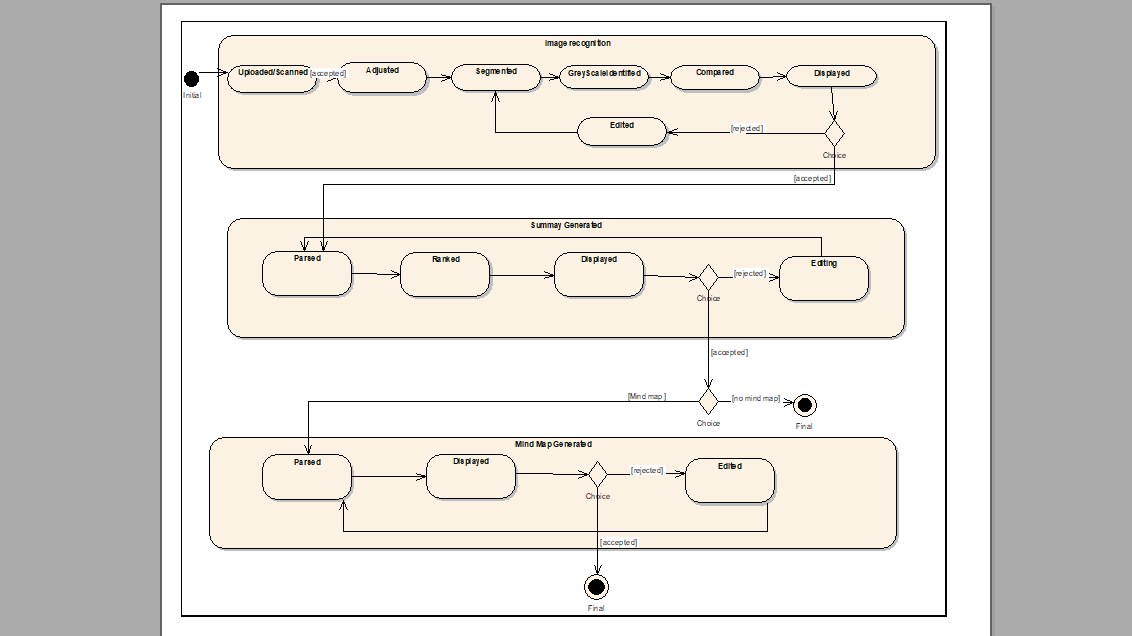
[](Class%20diag.png)

**4.2.3 Sequence Diagram:** A sequence diagram is interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

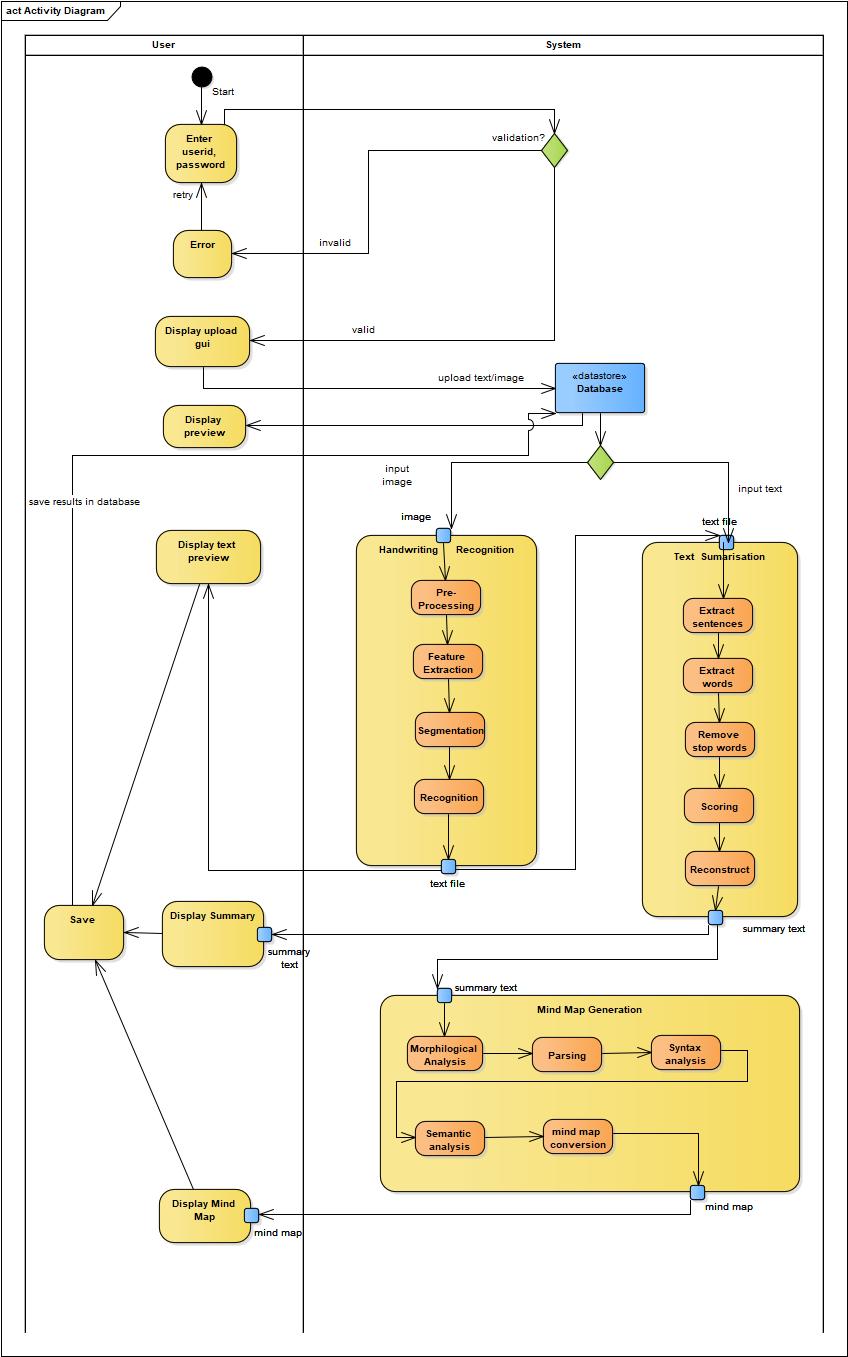
A sequence diagram shows object interaction arranged in time sequence. It depicts the object and classes involved in the scenario and the sequences of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

[](Sequence%20diagram.jpg)

**4.2.4 State Diagram:** A state diagram is a type of diagram used in coputer science and related fields to describes the behaviour of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is reasonable abstraction. Many forms of state diagrams exists, which differs slightly and have differebt semantics.

[](state.png)

**4.2.5 Activity Diagram:** Activity Diagram are graphical representation of workflows of stepwise activites and actions with support for choice, iteration concurrency. In the Unified Modeling Language(UML), acitivity diagrams are intended to model both computational and organizational processes(i.e workflows). Activity diagrams shows the overall flow of control.

[](Activity%20Diagram.jpg)

**DOCUMENT SUMMARIZATION**

**AND**

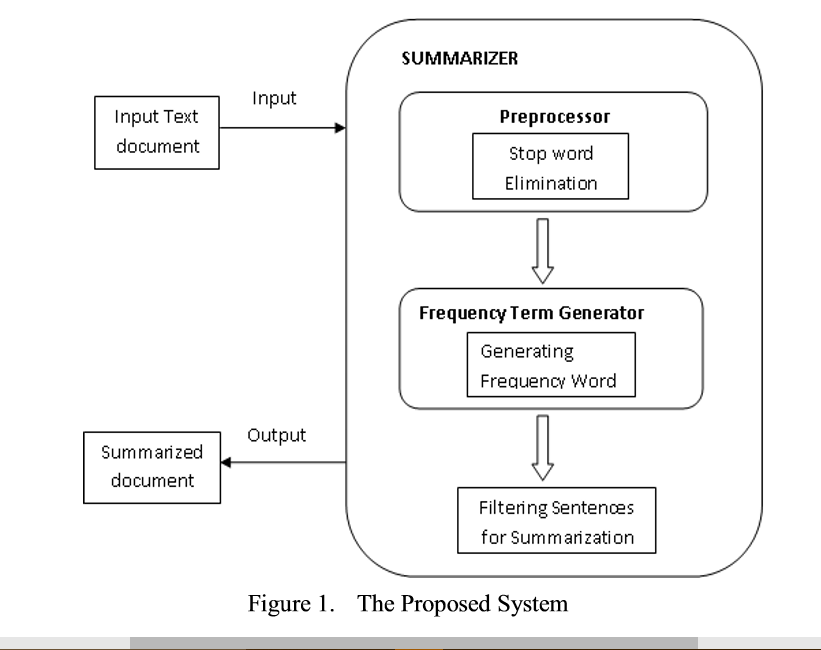
**MIND MAP**

**CHAPTER 5**

**ARCHITECTURE OF THE SYSTEM**

* + 1. **Architecture of Document Summarization:**

It describes how the text document or handwritten text document gets converted to a summarized document from the series of following steps. Firstly, the preprocessing of the document happens which eliminates the stop words present in it. Secondly, frequency of each word is calculated. And finally sentences with more frequency are included in the summarized document. Optionally, user can ask for mind map which can be generated from the summarized documents.



* + 1. **Architecture of Mind Map:**

**(put here desc and then architecture)**

**CHAPTER 5**

**RESULTS AND DISCUSSIONS**

**(include a chapter of testing and one of sample code)**

**CHAPTER 6**

**APPLICATIONS(REFINE IT)**

* Generation of Summarized Information from huge Textual Documents which can be used in the fields
* Education – Simpler Notes of Complex Study Materials
* Government Documents – Simplifying Complex government Documents.
* Library Management – Generating abstarct of every book for easy selection of required Book.
* Abstract Generation – of novels, news reports, magazines
* Title Generation
* Generation of Graphical representation – a mind map of summarized Documents
* Regonition of relationship between different objects
* Better and faster understanding of complex Information

**CHAPTER 7**

**CONCLUSION**

The conclusion of the project is that we developed an automated and time efficient software system using the most suitable algorithm for Text Summarizatio and Mind Map.

(REFER SIR FOR THIS)

**REFERNCES**

[1]. Moratanch, N., and S. Chitrakala. "A survey on extractive text summarization." Computer, Communication and Signal Processing (ICCCSP), 2017 International Conference on. IEEE, 2017.

[2]. Abdeen, M., et al. "Direct automatic generation of mind maps from text with M 2 Gen." Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference. IEEE, 2009.

[3]. Agarwal, Mohit, and Baijnath Kaushik. "Text recognition from image using artificial neural network and genetic algorithm." Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on. IEEE, 2015.

[4]. Krishnan, Praveen, Kartik Dutta, and C. V. Jawahar. "Deep feature embedding for accurate recognition and retrieval of handwritten text." Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on. IEEE, 2016.

[5]. Rabi, Mouhcine, et al. "Recognition of cursive Arabic handwritten text using embedded training based on HMMs." Engineering & MIS (ICEMIS), International Conference on. IEEE, 2016.