
Software Design Specification

For

**Web Usage
Mining to
understand
Consumer
Trends**

Version 1.0 approved

Prepared by -

Ayan Chandra

Amit Kumar

Buddhi Rohan Sai

Table of Contents

1. Introduction

1.1 Purpose

1.2 System Overview

2. Design Considerations

2.1 Assumptions

2.2 Constraints

2.3 System Environment

2.4 Risks and Volatile Areas

3. Architecture 2

3.1 Overview and Architectural Design Diagram

3.2 Subsystem, Component, or Module

3.3 Strategy

4. Design Diagrams

4.1 Use Case Diagram

4.2 Activity

4.3 Sequence

4.4 Collaboration

4.5 Component

4.6 State

4.7 Class

4.8 Object

4.9 Deployment

1. Introduction

1.1 Purpose

This design will detail the implementation of the requirements as defined in the Software Requirements Specification – Web Usage Mining to understand Consumer Trends.

1.2 System Overview

The purpose of this project is to enable anyone to easily analyze the current consumer trends, i.e, how people are consuming the internet, and look into people's sentiments regarding it. This is useful in knowing how the people online feel about a certain incident or a product or a company or anything which is trending. This may also help small-business owners and small-time investors to check the general sentiment and trend of their investments.

2. Design Considerations

2.1 Assumptions

- Users have access to internet
- Users have a compatible web browser
- Users have devices which can connect to the internet

2.2 Constraints

None that we are aware of.

2.3 System Environment

This Project is a website and hence doesn't depend on any Operating System and works in any modern browser that can run HTML5, as long as an internet connection is available.

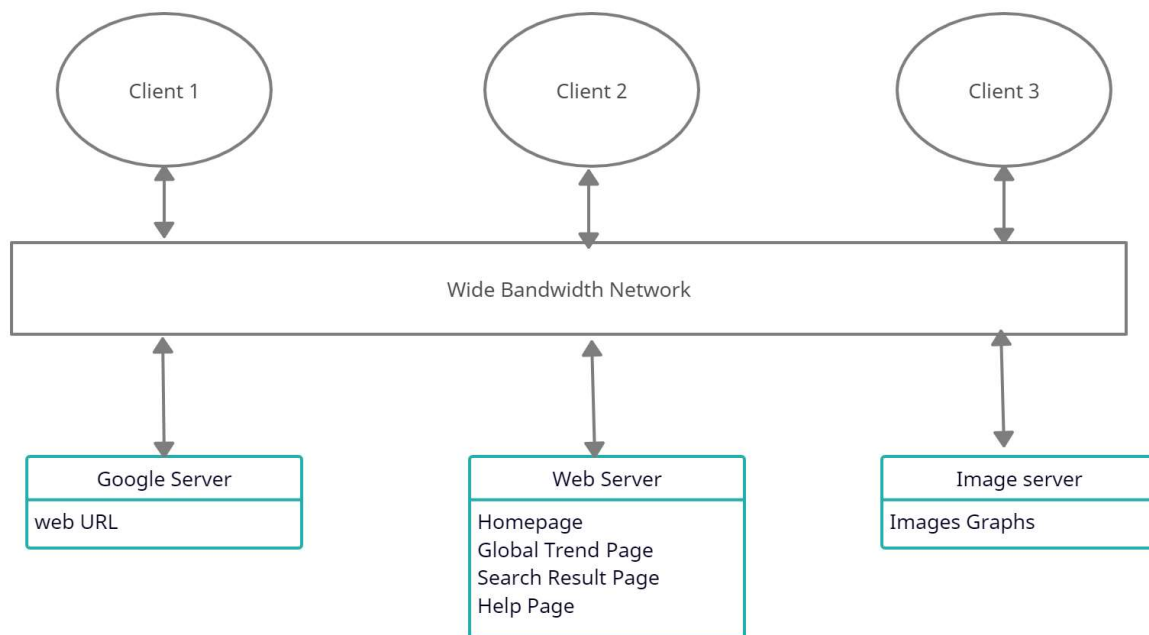
2.4 Risks and Volatile Areas

- User Information should not be stored unknowingly
- Users should not be allowed to tamper with the website in any manner.

3. Architecture

3.1 Overview

The most appropriate architecture model for this project will be the client-server model because the website will be hosted on a remote server and the users will have to access it over a wide-bandwidth network.



3.2 Modules

1. **Home page:** The first page shows the worldwide trends and shows the sentiment of the given trends.

viewHome() and disphome() are used so that the website can deploy the homepage and the user can view it.

2. **Trend visualization:** The worldwide trends will be visualized on a globe in a visual manner.

visualizetrend() and displaytrendgraph() helps in order to plot graphical representation of the data collected and display it to the user on the page.

3. **Named entity recognition:** The process will be implemented to scrape data from the web and find the usage of the data to hence find the trends of the topics by using nltk.

4. **Sentiment analysis:** The sentiment on the trends will be found out by using nltk and scraping various social media sources for opinions.

viewsentiment() and returnsentiment() help in order to analyze the trends and graphical representation and provide a sentiment according to it to the user

5. **NER for region:** The region based trends will have to be found by using the NER and by filtering it according to the specified region.

6. **Region-based trend page:** This page will allow us to select the region and display the trends for that specific region.

7. **Search module:** A particular keyword's trends can be found by searching for it along with its sentiment.

search() and returnsearch() help to search the trends by the text input we provide and then according to the regions it is sorted, and displayed to the user who can go ahead with the search results or can filter regions according to his/her choice.

8. **Search page:** This page will display the results of the searched keyword's trends and sentiment.

returnsearch() helps in displaying the results of what trend we search for.

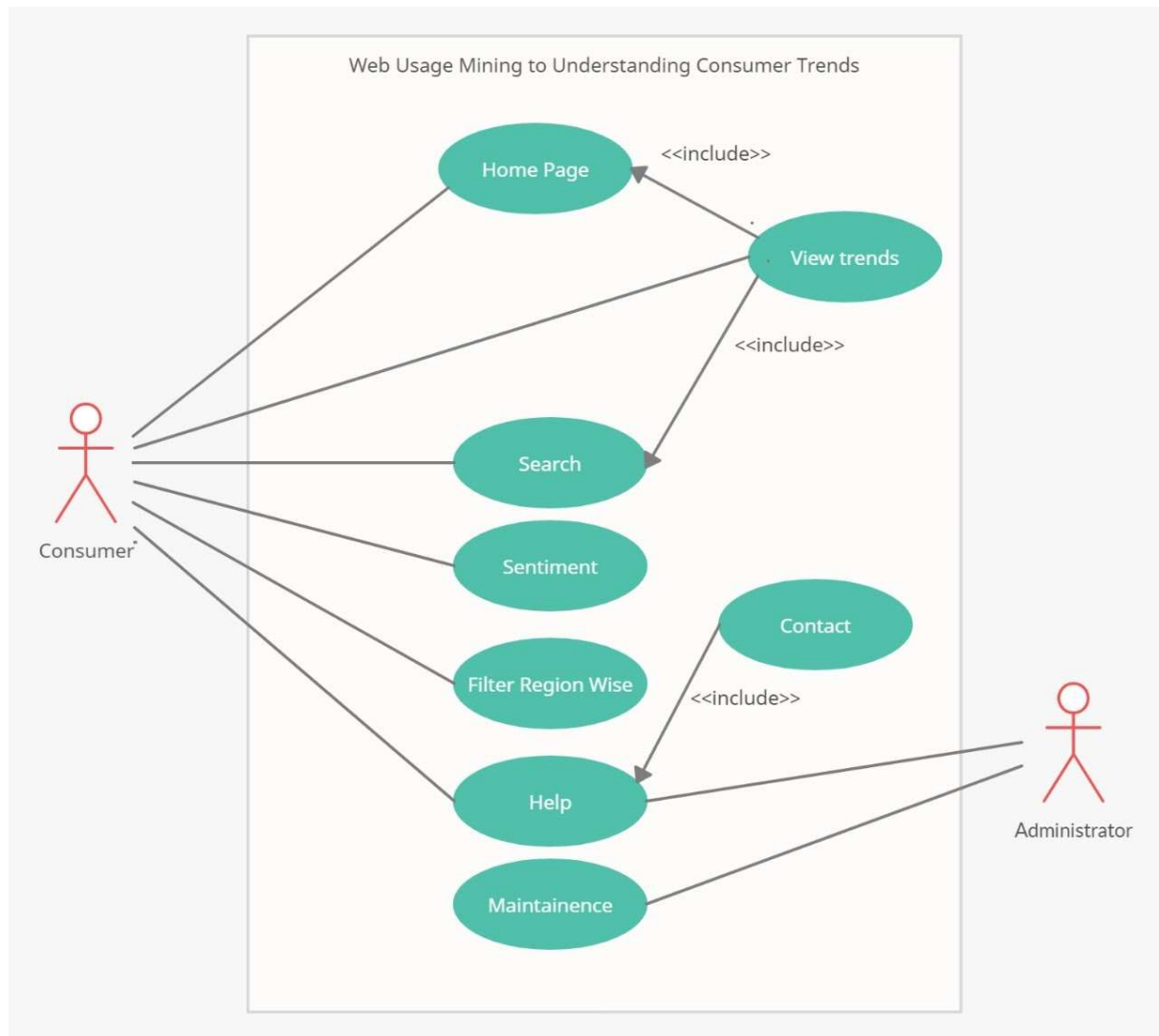
9. **Website:** All the pages will then be integrated into a website.

3.3 Strategy

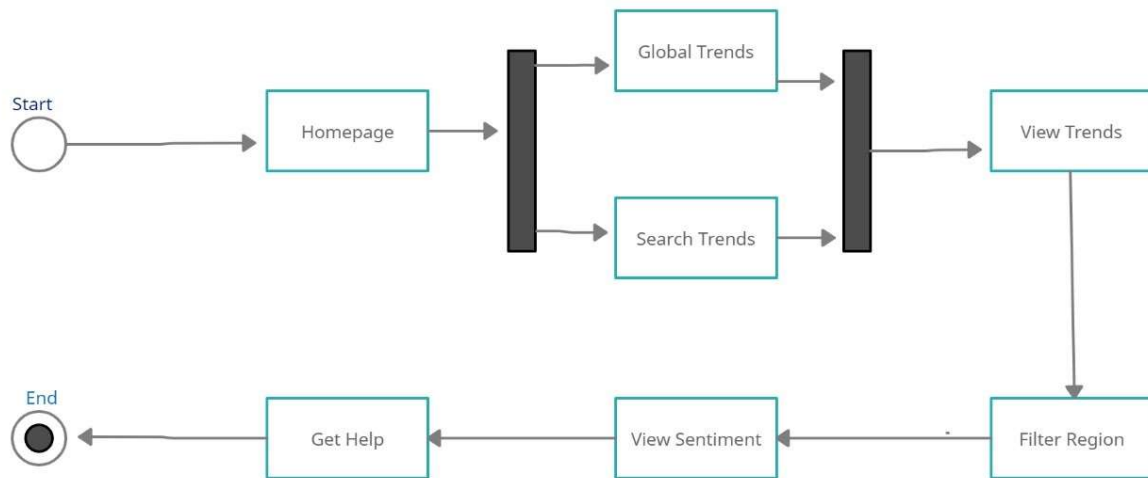
- It's a design philosophy that breaks down an application or website into discrete, reusable, and self-contained objects.
- In an object-oriented system, the amount of work involved in revising and maintaining the system is reduced, since many problems can be detected and corrected in the design phase.
- It helps in the faster development of software.
- It allows for relatively painless upgrades.
- It is easy to maintain. Suppose a module develops an error, then a programmer can fix that particular module, while the other parts of the software are still up and running.
- It enables the reuse of objects, designs, and functions.

4. Design Diagrams

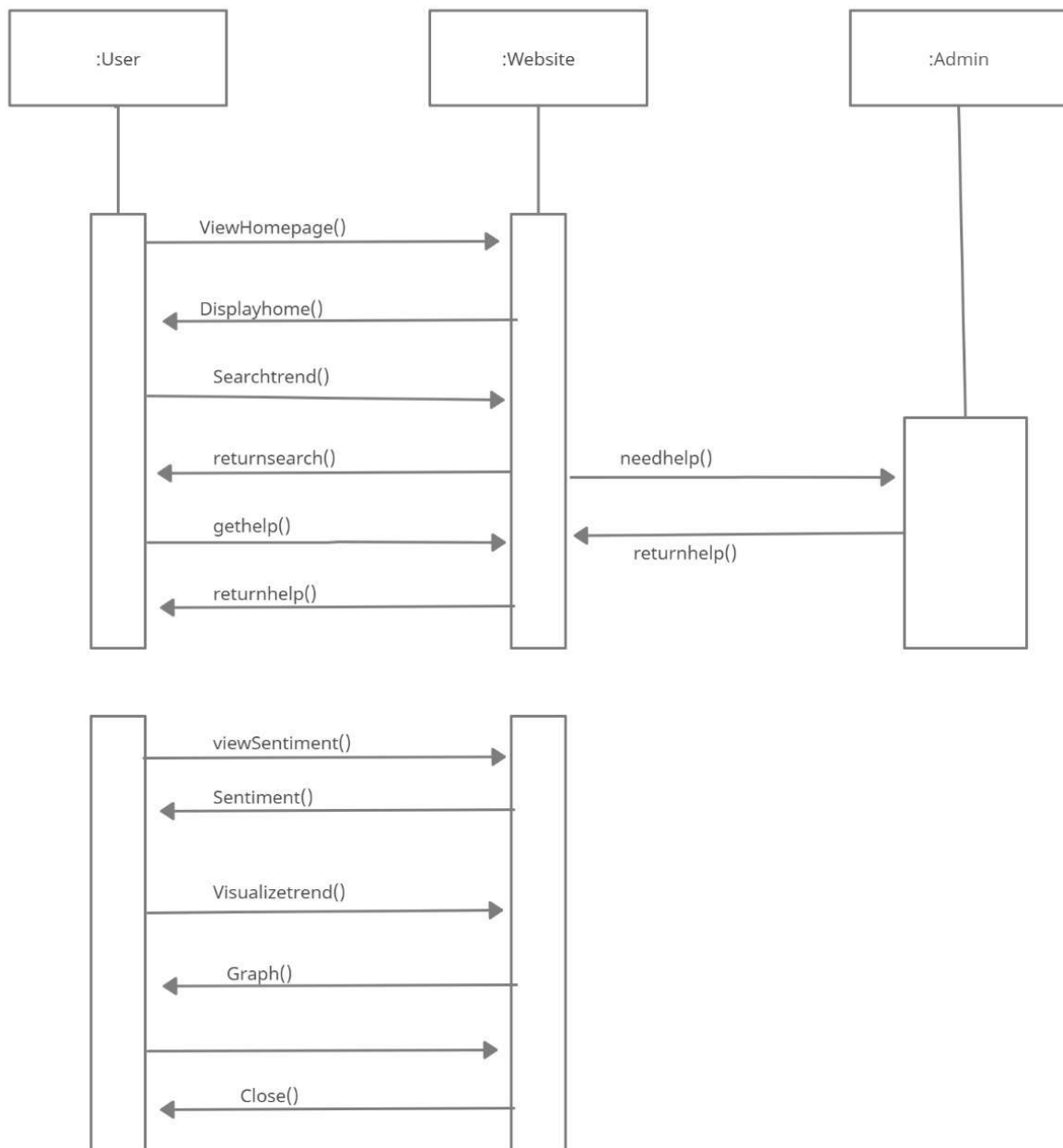
4.1 Use Case Diagram



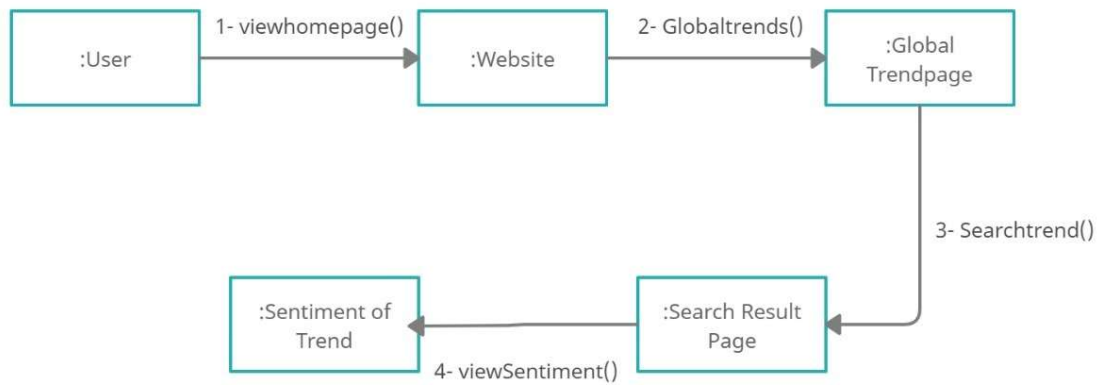
4.2 Activity Diagram



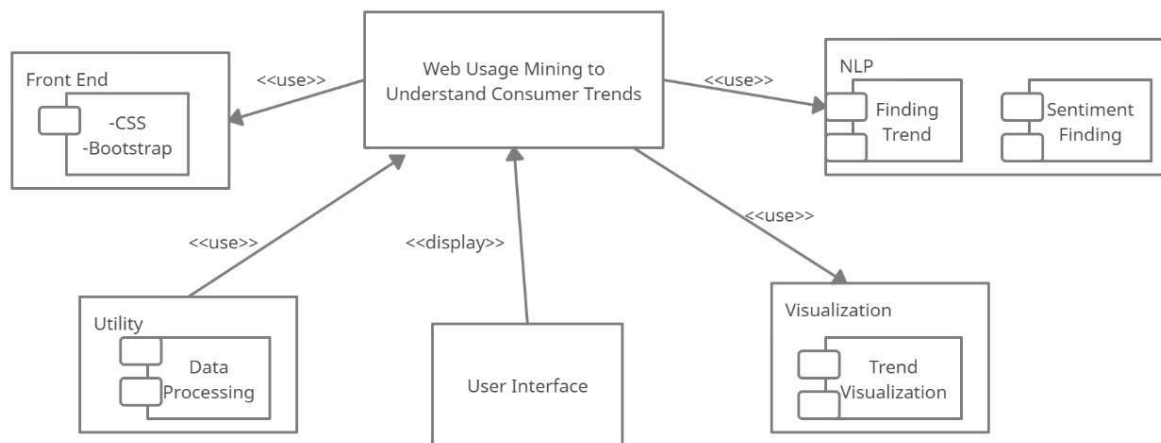
4.3 Sequence Diagram



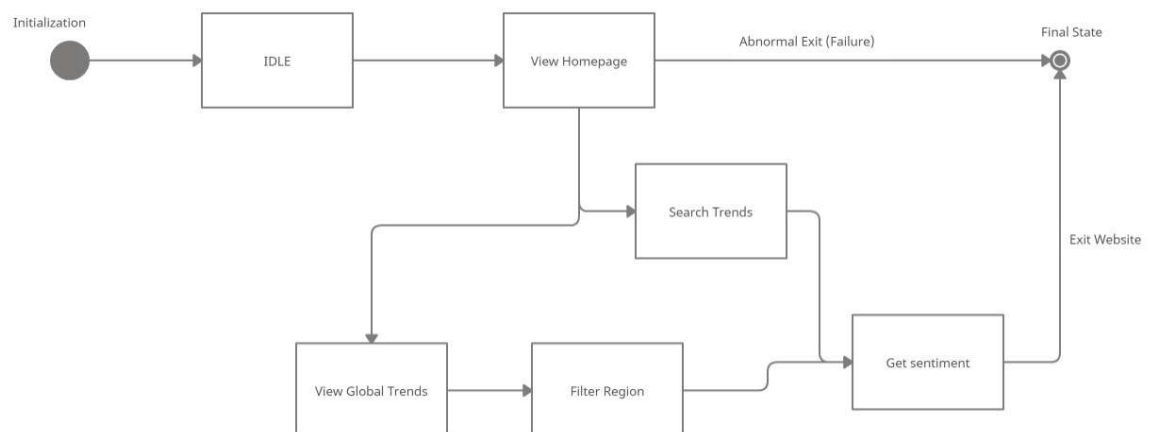
4.4 Collaboration Diagram



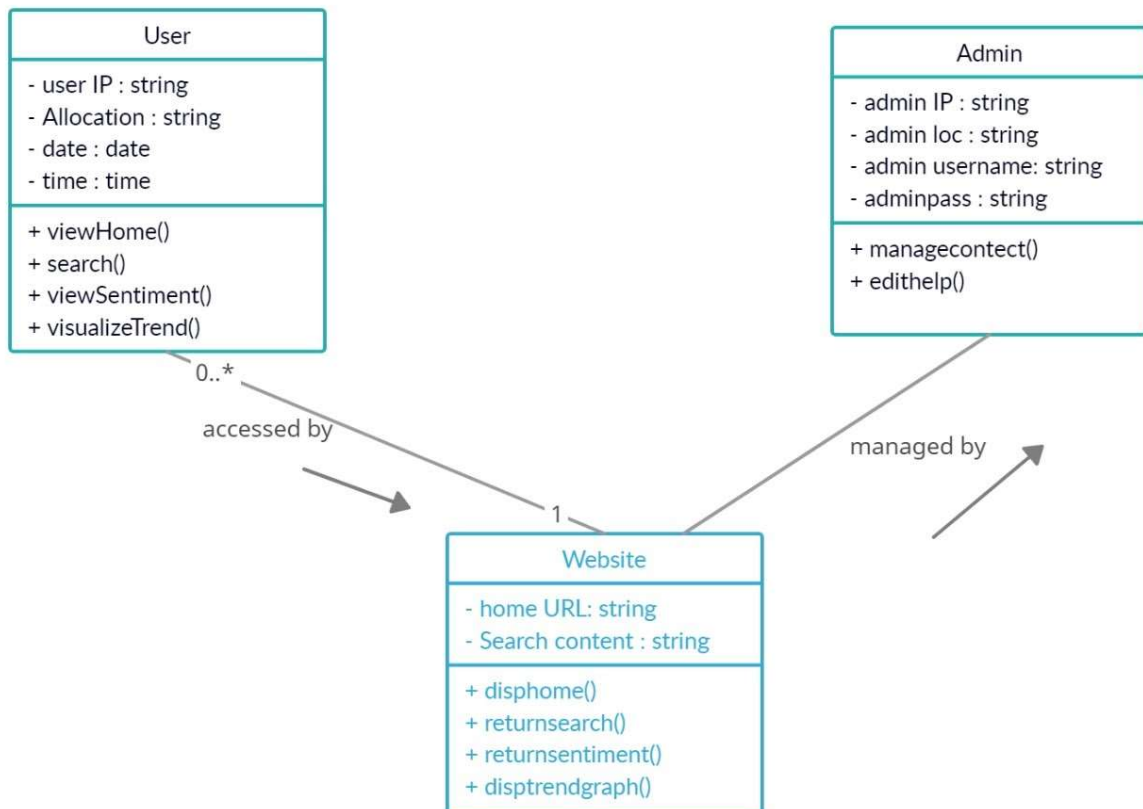
4.5 Component Diagram



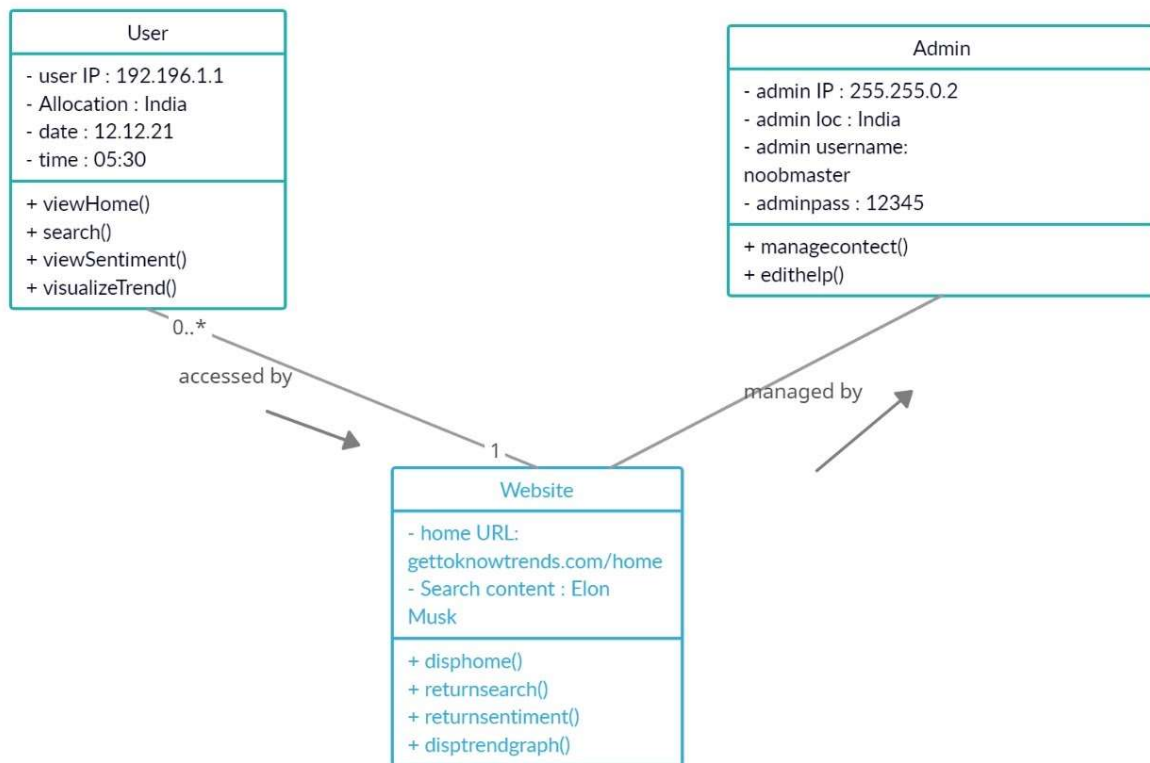
4.6 State Chart Diagram



4.7 Class Diagram



4.8 Object Diagram



4.9 Deployment Diagram

