

SOFTWARE DESIGN DOCUMENT

SÜZGEÇ TURKISH TEXT SUMMARIZATION

SUPERVISORS

Ayşenur BİRTURK
İtir ÖNAL

Team Default

Abdullah Göktuğ MERT
Baran Barış KIVILCIM
Enes Uğur ŞEKERCİ
Yağız ARKAYIN

Table of Content

1. Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 Definitions, Terms and Abbreviations	2
2. References	3
3. Design Viewpoints	3
3.1 Context Viewpoint	3
3.1.1 Use Case 1: Summarize Web Page with Chrome Extension	3
3.1.2 Use Case 2: Summarize Selected Text with Chrome Extension	4
3.1.3 Use Case 3: Summarize Web Page with Web Application	5
3.1.4 Use Case 4: Summarize Text with Web Application	6
3.1.5 Use Case 5: Summarize File with Web Application	7
3.1.6 Use Case 6: Change Summary Length Settings	8
3.2 Composition Viewpoint	8
3.3 Logical Viewpoint	10
3.3.1 Text Class	10
3.3.2 Sentence Class	11
3.3.3 Word Class	11
3.4 Interaction Viewpoint	11
3.4.1 Web Application Interactions	11
3.4.2 Chrome Extension Interactions	12

1. Introduction

The Software Design Document is a document which provides the details for how the intended software should be built. The details are represented by using graphical notations such as use case models, sequence diagrams, class diagrams, object behaviour models, and other supporting requirement information.

1.1 Purpose

The purpose of this document is to describe and visualize the architecture of Turkish Text Summarizer Süzgeç by different viewpoints.

This document aims to describe the software system which is structured to meet the needs specified in Software Requirements Specification for the implementation phase.

The expected audience for this document is the design and development team of the Süzgeç Project for implementing purposes. Besides team leader, supervisor and CENG490 lecturers use this document for reviewing purposes.

1.2 Scope

This document contains a structural overview of all modules, interfaces and data. It also covers a detailed design of each module by giving information about the overall software architecture and the design methods for each module of the software product. A set of design views will be presented in order to support the design and development process. This document will serve as a guideline through the implementation phase.

1.3 Definitions, Terms and Abbreviations

User: A person who uses either web application or chrome extension services.

Cloud Service: Any resource that is provided over the Internet

Chrome Extension: A plug-in that extends the functionality of Google Chrome Browser.

Server: A computer or computer program which manages access to a centralized resource or service in a network.

Summarization: Sum up the main points of something, in this case: selection of important sentences from the text.

2. References

[1]IEEE STD 1016-2009, IEEE Standard for Information Technology-Systems Design- Software Design Descriptions

3. Design Viewpoints

3.1 Context Viewpoint

The context view point describes the relationships, dependencies, and interactions between the system and its environment.

The following use case diagram are presented to show an overview of the relationships and interactions between actors and the system.

3.1.1 Use Case 1: Summarize Web Page with Chrome Extension



Figure 1 : Use Case Diagram 1

Use Case ID	1
Use Case	Summarize Web Page with Chrome Extension
Description	Getting summary of a webpage without any selections on the webpage
Actor	User
Trigger	User clicks "Web Sayfasını Özetle" button in the extension pop-up.

Primary Scenario	<ul style="list-style-type: none"> - User clicks “Web Sayfasını Özetle” button. - System takes the web page and sends it to the cloud server. - Web page is parsed, and the summarization algorithm takes its body part as input. - The algorithm ,then, sends the summary sentences to the extension. - The extension show the summary of the website.
Exceptional Scenario	None

3.1.2 Use Case 2: Summarize Selected Text with Chrome Extension



Figure 2 : Use Case Diagram 2

Use Case ID	2
Use Case	Summarize Selected Text with Chrome Extension
Description	Getting a summary of the webpage but only the selected part.
Actor	User
Trigger	<ul style="list-style-type: none"> - User selects some parts of the text on the webpage, by using the cursor. - User clicks the “Web Sayfasını Özetle” button on the extension
Primary Scenario	<ul style="list-style-type: none"> - User clicks “Web Sayfasını Özetle” button. - System takes the sentences that user selected. - Extension sends these sentences to the cloud server. - The algorithm running on the cloud takes these sentences as input and produces summary sentences. - Summary sentences are then sent back to the extension
Exceptional Scenario	None

3.1.3 Use Case 3: Summarize Web Page with Web Application



Figure 3 : Use Case Diagram 3

Use Case ID	3
Use Case	Summarize Web Page with Web Application
Description	Getting a summary of a webpage with providing its link to the server.
Actor	User
Trigger	<ul style="list-style-type: none">- User obtains the URL of the webpage whose summary he/she wants- User directs to the webpage of “Süzgeç” and supplies the link on “Link giriniz” section and clicks “özetle” button
Primary Scenario	<ul style="list-style-type: none">- User clicks “özetle” button.- System takes the URL and parse that webpage.- On the cloud the algorithm is run and summary sentences are produced.- The sentences are then presented on the webpage’s summary section.
Exceptional Scenario	None

3.1.4 Use Case 4: Summarize Text with Web Application



Figure 4 : Use Case Diagram 4

Use Case ID	4
Use Case	Summarize Text with Web Application
Description	Getting a summary of user entered text
Actor	User
Trigger	<ul style="list-style-type: none">- User enters or copies some text to websites required text area.- Then, on the website, clicks the “Özetle” button
Primary Scenario	<ul style="list-style-type: none">- User clicks “Özetle” button.- System takes the user entered text.- The algorithm runs and produces the summary sentences.- These sentences are sent to website and shown in the summary text area.
Exceptional Scenario	None

3.1.5 Use Case 5: Summarize File with Web Application

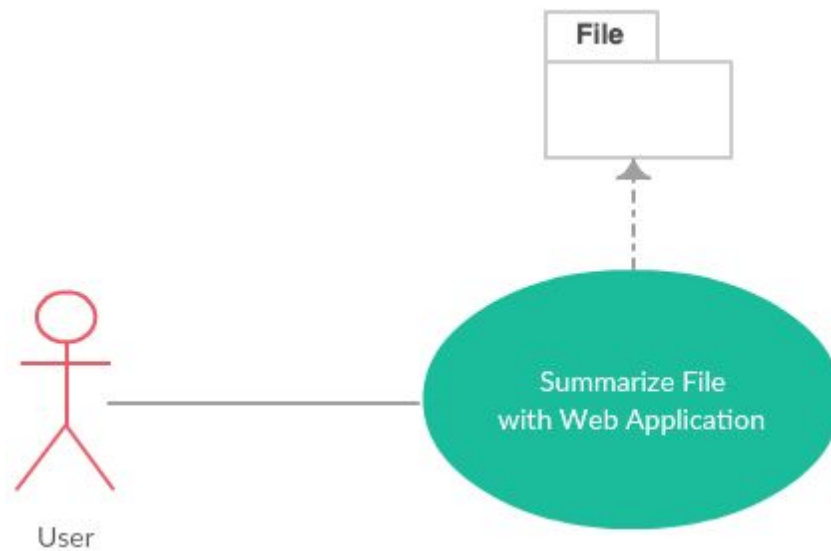


Figure 5 : Use Case Diagram 5

Use Case ID	5
Use Case	Summarize File with Web Application
Description	Getting a summary of a text file
Actor	User
Trigger	<ul style="list-style-type: none"> - Using the website the user uploads a file to the system, then clicks the “özetle” button.
Primary Scenario	<ul style="list-style-type: none"> - User clicks “Özetle” button after uploading a text file - The server if necessary converts the text file and then parse the final file. - The parsed sentences are used by the algorithm to produce the summary sentences. - The summary sentences are then given to the website to be shown.
Exceptional Scenario	None

3.1.6 Use Case 6: Change Summary Length Settings



Figure 6 : Use Case Diagram 6

Use Case ID	6
Use Case	Change Summary Length Settings
Description	Deciding how many sentences will be in the summary.
Actor	User
Trigger	<ul style="list-style-type: none">- User goes in the settings and then click the button to determine the length of the summary.- User continues using the service with this setting
Primary Scenario	<ul style="list-style-type: none">- User clicks the button to change the length of the summary- Supplying text and clicking the “özetle” button user sends the text to be summarized.- After getting the text, the server updates the settings of the algorithm.- The algorithm runs with the updated settings and produces summary as output.- The summary sentences then sent back to the website/extension.
Exceptional Scenario	None

3.2 Composition Viewpoint

This viewpoint focuses on the structure of the system and provides a top level view of the entire system from the perspective of each component.

For this aim, for the logical representation UML Component Diagram and for the physical representation UML Deployment Diagram is provided that can be seen as below;

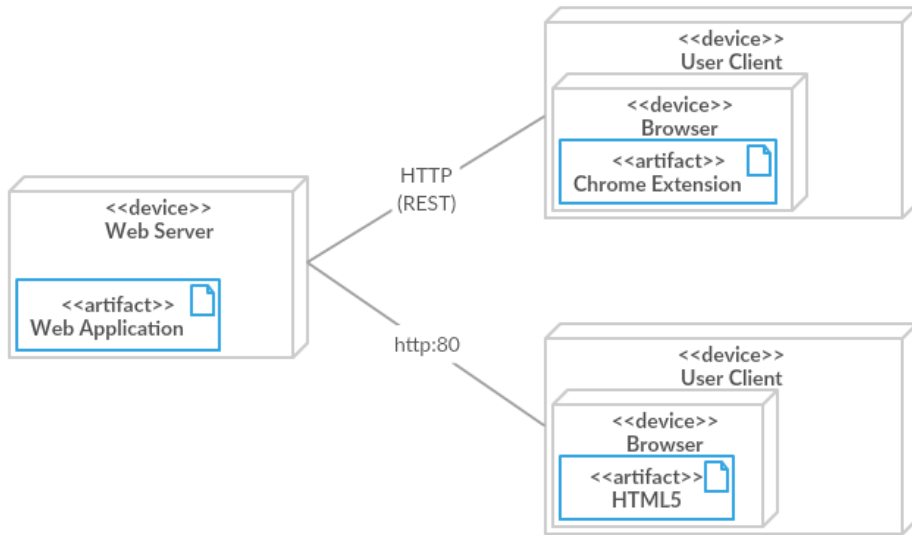


Figure 7 : Deployment Diagram

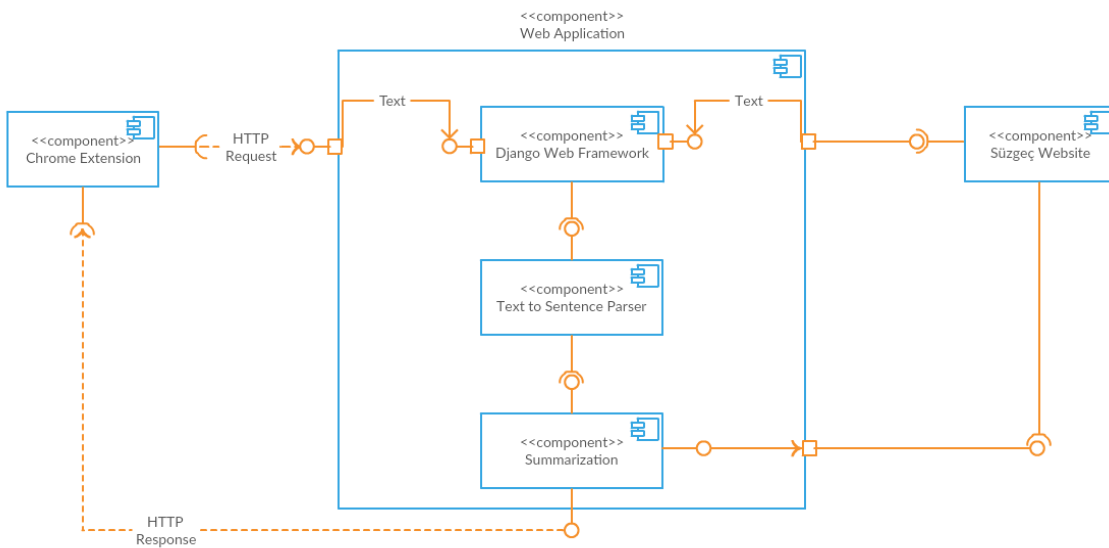


Figure 8 : Component Diagram

3.3 Logical Viewpoint

This viewpoint aims to show the key abstractions such as classes and interactions among them. UML Class diagram is provided for this aim which can be seen as below.

There are 4 main classes in our project: Text, paragraph, sentence and word. Each of their fields and methods are going to be used in order to calculate the sentence features.

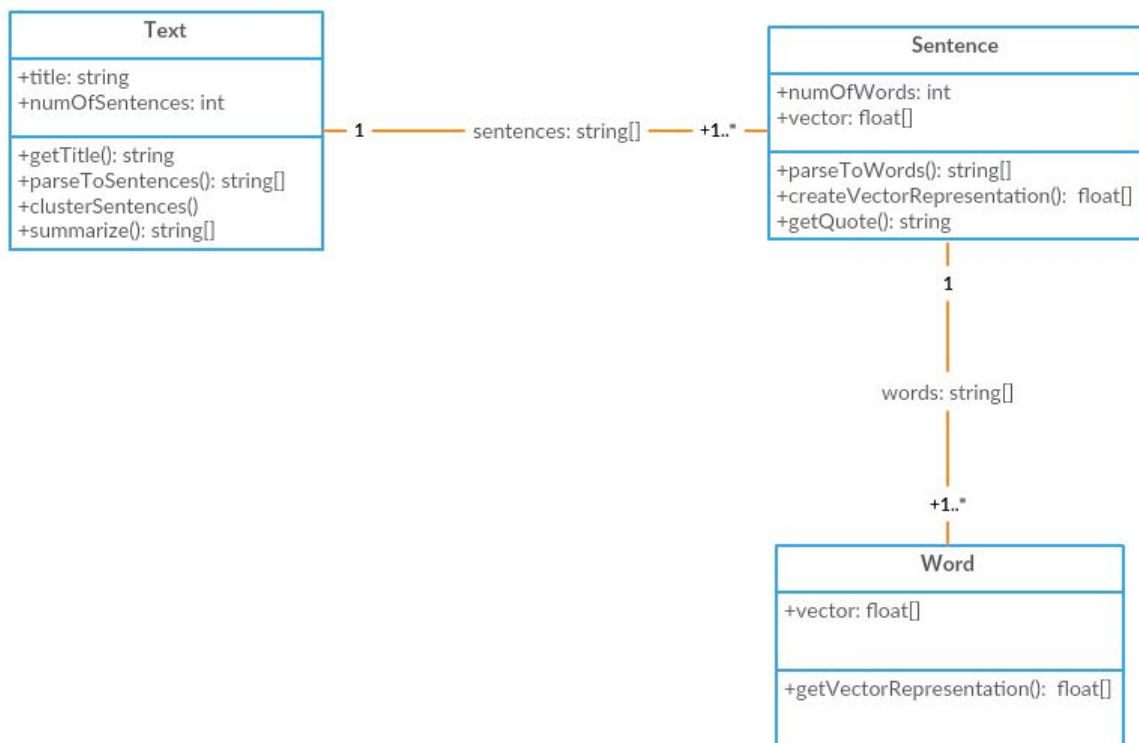


Figure 9 : Class Diagram

3.3.1 Text Class

Text class is the most complex class of the system. It has sentences and words. For dividing text to these parts, text class should have parser methods. Also there are title and number of sentences attributes in this class.

“summarize” method uses “clusterSentences” method to cluster sentences, and from these clusters summary sentences are obtained.

3.3.2 Sentence Class

Sentence class is the most important class of the system. It has vector representation attribute as a float list and “numOfWord” attribute. There is a parser function to parse sentence to words. Sentence object has a method to calculate vector representation of itself with the vector representation of its words. It has a float list called “vector”. “vector” list has 200 float values that is obtained from words’ vectors summation with “createVectorRepresentation” method.

3.3.3 Word Class

Word class is the most basic class of the system. Using Word2Vec, it contains vector representation that consist of 200 float values. Also it has a method that can be used for getting its vector representation.

3.4 Interaction Viewpoint

In this view point the interaction the interaction among entities of the system will be visualized. UML Sequence Diagrams are used in order to provide representation of the interaction.

3.4.1 Web Application Interactions

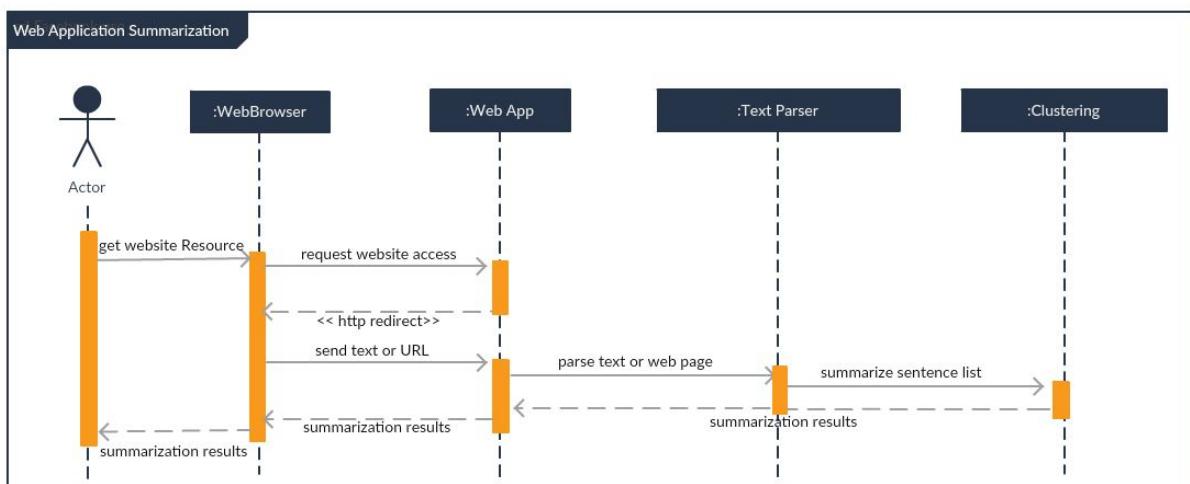


Figure 10 : Web Application Sequence Diagram

User accesses the web application over his/her web browser. Then, he/she enters either a text or URL of a website whose content will be summarized. Web application directs this text or URL to the text parser. The text parser parses through the text and detect the sentences. This sentences are then sent to clustering algorithm where the points of each sentence in space is found and later these points are clustered. The clustering algorithm returns the summary sentences all the way back to the user.

3.4.2 Chrome Extension Interactions

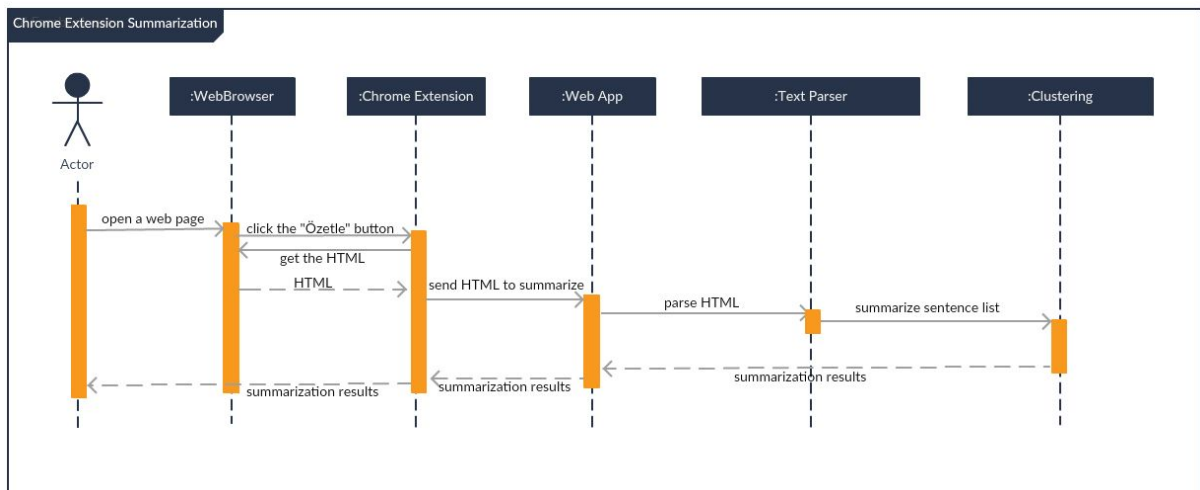


Figure 11 : Chrome Extension Sequence Diagram

The diagram shows how the Chrome Extension works when the summarization process is in action. First, the user clicks the "özetle" button when browsing a webpage. This click triggers the extension. The extension sends the HTML contents of the webpage to the web application. The web application parses the HTML and obtains the necessary text parts of the website, excluding the advertisements and other unnecessary parts. After the text is obtained, a list of sentences is sent to the clustering algorithm. Using these sentences as input, the clustering algorithm finds the summary sentences among them. Finally, these summary sentences are displayed on the extension.