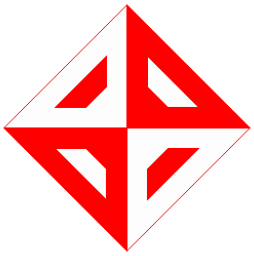

METU - Department of Computer Engineering

SOFTWARE DESIGN DOCUMENT

WebCat

3 June 2016

Group Name :

Overcode

Group Members:

Mert Basmacı - 1819119

Onur Ozan Yüksel - 1881663

İzzet Barış Öztürk - 1881796

Özge Donmaz - 1819234

CONTENTS

1. Introduction	1
1.1. Purpose	1
1.2. Scope	2
1.3. Overview	2
2. Definitions, Acronyms and Abbreviations	2
3. Design Viewpoints	2
3.1. Context Viewpoint	2
3.2. Composition Viewpoint	8
3.3. Dependency Viewpoint	9

1. Introduction

1.1. Purpose

This software design document is intended to give the reader a clear perspective about WebCat, which is the first web categorization tool designed for Turkish web pages. The document is to describe the functional structure of the system, also the system resources. The intended audience of this project could be anyone who seeks web page categorization or web page filtering; however, it’s been predicted that the companies who need filtering for their networks during the working hours has the lion's share of the intended audience.

1.2. Scope

The finalized version of the system can identify the category of a web page by going through it's written content. There are 125 category alternatives into which Turkish web pages can be grouped. By this way, the system has also a pioneer role (being the first Turkish web-categorizer aside) because this much alternative hasn't been tried before. It is also given importance to systems performance that enables users to perform quick categorization.

1.3. Overview

The system is composed of different components such as web-crawler, database and machine learning tools and algorithms. Data is collected on the background by the web-crawler. All the Turkish web pages are visited, categorized and recategorized (if needed) into groups regularly. Categorization is done by machine learning having the supervised and semi-supervised learning sense. The system takes the advantage of using database in crawling and categorizing phases due to that it performs operations on big data.

2. Definitions, Acronyms and Abbreviations

IEEE	The Institute of Electrics and Electronics Engineers
TCP	Transmission Control Protocol
Weka	Waikato Environment for Knowledge Analysis
JDBC	Java Database Connectivity

3. Design Viewpoints

3.1. Context Viewpoint

This section is aimed to denote the interaction between the system and the users of the system. Use case and interaction-overview diagrams are used to depict those interactions. You can find the diagrams below. (See Figure 1-2-3 and their explanations)

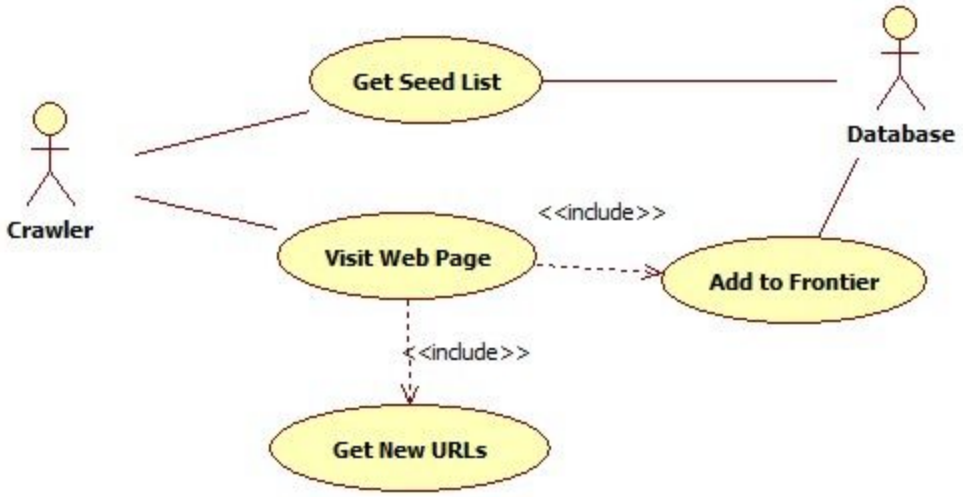

Figure 1: Use Case Diagram 1

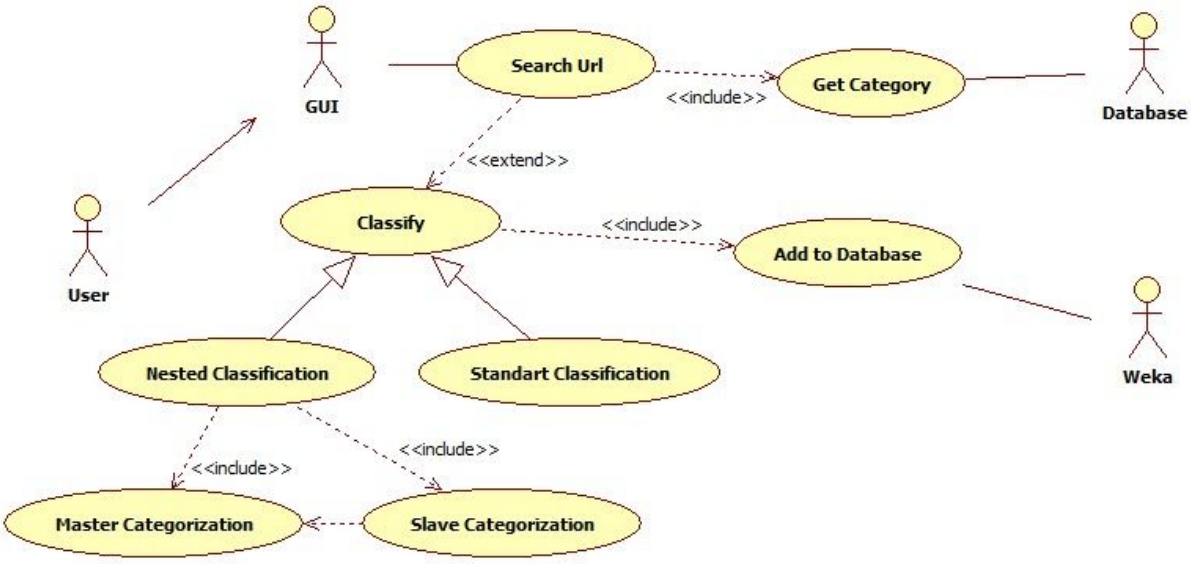

Figure 2: Use Case Diagram 2

ID:	UC-1
Title:	Already Categorized
Description:	The URL that user entered is already categorized and stored in database.
Primary Actor:	User / GUI
Preconditions:	URL and it's category is in database
Postconditions:	The category is returned to user
Main Success Scenario:	1. User types the URL of the webpage into the space on GUI page. 2. User presses “Enter” button or clicks “Categorize” button. 3. GUI connects to Database. 4. Makes the category requesting query. 5. Database returns the category of the page. 6. GUI shows the category in the text box just below the URL.
Extensions:	3a. GUI can't connect to Database. Shows connection error on screen. 5a. Database returns category as “Not Found” if the condition wasn't met. Refer to UC-2.
Frequency of Use:	Half of the queries. (probably)

ID:	UC-2
Title:	Needs to be Categorized
Description:	The URL that user entered was not categorized at the time user enters.
Primary Actor:	User / GUI
Preconditions:	URL and it's category is not in database
Postconditions:	Web page categorized and its added to the database. GUI returns its category.
Main Success Scenario:	1. User types the URL of the webpage into the space on GUI page. 2. User presses “Enter” button or clicks “Categorize” button. 3. GUI connects to Database. 4. Makes the category requesting query. 5. Database returns the empty list to the query. 6. GUI asks user the desired classification method.(Standard or Nested) 7a. If user has chosen “Standard Classification”, standard classification function is called and the function returns it's category in few seconds. 7b. If user has chosen “Nested Classification”, master classification function is called then the slave classification function is called with the output of the master classification function. The slave classification function returns the category of the web page. 8. The newly found category is added to the database alongside with the URL. 9. Found category is shown in GUI.
Extensions:	3a. GUI can't connect to Database. Shows connection error on screen.
Frequency of Use:	Half of the queries. (probably)

ID:	UC-3
Title:	Crawler Initial Start
Description:	The crawler starts and continues indefinitely.
Primary Actor:	Crawler
Preconditions:	Seedlist should be filled in database.
Postconditions:	-
Main Success Scenario:	1. The crawler program starts. 2. Crawler gets its initial web pages to visit list from database. 3. Visits the next page on the list. 4. Stores the page's url in to-be-categorized table in database. 5. Finds the new links from the page and adds them to to-visit list. 6. Repeats from 3rd step till the list finishes. 7. Repeats from 2nd step.
Extensions:	2a. Database connection fails shows error and stops. 2b. The seedlist table in database is empty, precondition was not met. Shows error and quits.
Frequency of Use:	1 time at initial start.

ID:	UC-4
Title:	Categorizer
Description:	The categorizer component that categorizes the pages that crawler had found.
Primary Actor:	Categorizer
Preconditions:	Crawler should be working before hand.
Postconditions:	-
Main Success Scenario:	1. The categorizer program starts. 2. Gets to-be-categorized list from the database. 3. Gets the next url on the list and confirms that it is not classified and stored in the database yet. 4. Gets the content of the web page belongs to the url. 5. Categorizes the content using standart classification. 6. Adds url and the newly found category to the database. 7. Removes the row from the database table that is being filled by crawler. 8. Repeats from 3rd step until the list is finished. 9. Repeats from 2nd step until the query returns empty. 10. Sleeps for 10 minutes. 11. Repeats from 2nd step.
Extensions:	2a. Database connection fails shows error and stops. 3a. The web page is already categorized and exists in the database, the program skips to 7th step then continues.
Frequency of Use:	1 time at initial start.

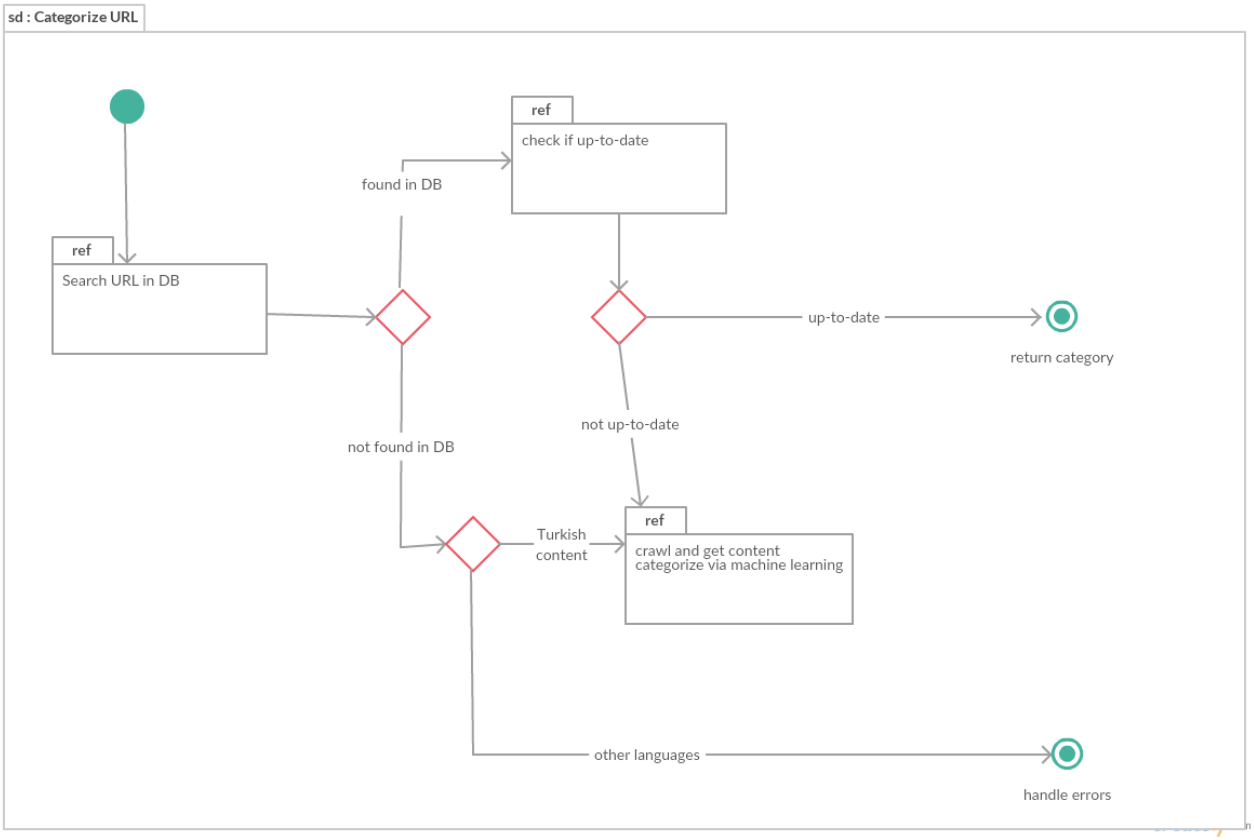

Figure 3 : Interaction-Overview Diagram

3.2. Composition Viewpoint

This section is intended to give a high level overview of how responsibilities of the system were partitions and assign into subsystems, namely GUI, web-crawler, database, machine learning algorithms and tools and their interactions between each other.

As you see in *Figure 4* below, there are three main categorization models stored in the system, slave, master and standard models respectively. Standard and nested classification are directly connected to the previously categorized data, where the model training is connected to machine learning component, textToArff, where supervised learning takes place. Standard and nested classification is used when a web page is queried. The database is checked if the corresponding web page is previously categorized, if not it is crawled and categorized. Crawler benefits from language detection and a built-in, but specialized crawler; machine learning component uses a built-in Turkish morphology tool as a subcomponent.

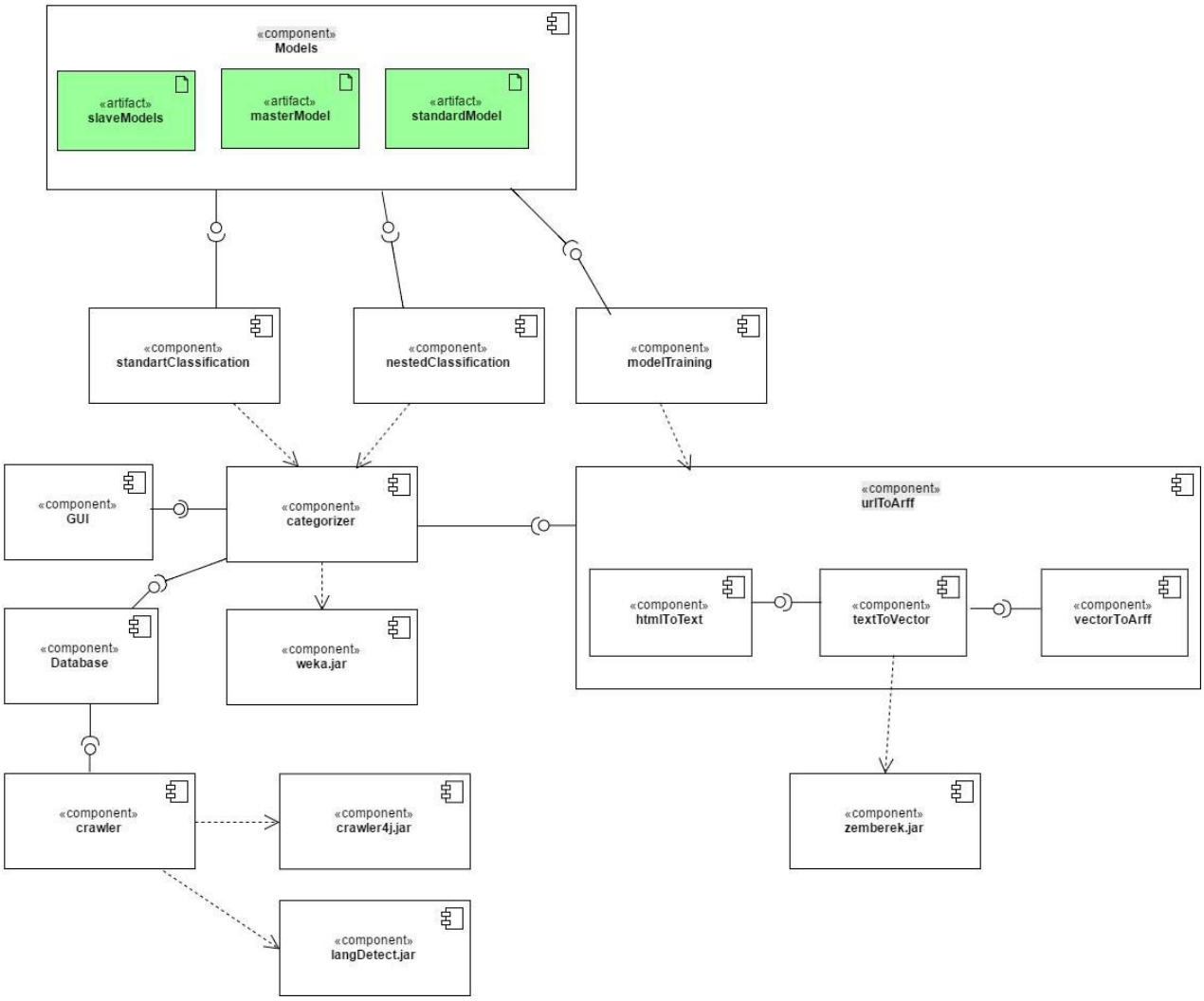

Figure 4 : Component Diagram

3.3. Dependency and Deployment Viewpoint

This section describes how the data flow across the components of the system. Dependencies between the components are depicted with deployment diagrams (see Figure 5).

As you can see from the deployment diagram below, the crawler and machine learning are put together to the application server when deployed. Data repositories and GUI are deployed into different servers, DB server and Web Server respectively. The systems is mainly design for API, but it is also reachable for other users from the Internet. The security and reliability is provided by using protocols between the servers (HTTP, TCP and JDBC). After the deployment process is done, system will be available with 90% guarantee.

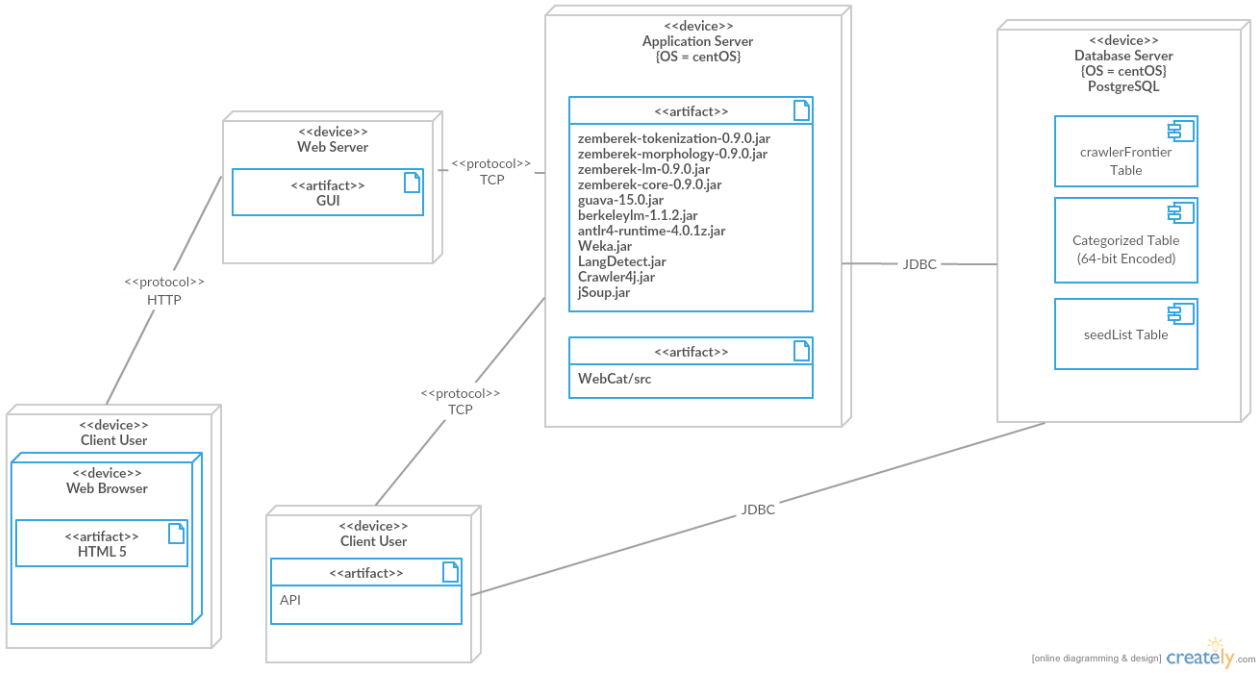

Figure 5 : Deployment Diagram