

What Am I Eating?

- Users who want to take care of their diets due to any disease or their preferences can use this application.
- After entering the application, users select the ingredients they want to eat or do not want eat.
- When users go to a restaurant; by taking a picture of the menu, they can see the meals in the menu match their preferences.

1-) Abstract

I've developed an android application that identifies food from restaurant menu with the help of phone camera, finds ingredients from the web service that I've implemented, and recommends food to users according to the user preferences.

The application allows the user to select food according to the materials he/she wants or does not want. In this way it is aimed to find the most suitable food for their health or diet.

2-) Introduction

In this project, I tried to find the most suitable meals according to the user preferences. It requires identifying meals' names from the restaurant menu and performing search over the recipes of the meals in order to give recommendation. Thus, I can say that the project consists of two main parts: Optical Character Recognition and Information Retrieval.

I've used Google's OCR engine for getting meals' names from the restaurant menu. I also used Lucene Library to create dataset and perform search over this dataset in order to give recommendation.

To create dataset, I've used some of the well-known meal websites that have recipes of the various types of meals. Then parsed the content of the web page to get information about the recipes by using Jsoup Library.

This report includes technical details about implementation of the used technologies and architectural design of the project. Also, the results of main scenarios and comments about development process are included.

3 -) Background

OCR(Optical Character Recognition): OCR is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document or a photo of a document. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line and used in machine processes such as cognitive computing, machine translation, text-to-

speech, key data and text mining. OCR is field of research in pattern recognition, artificial intelligence and computer vision.

I've used OCR to read meals in the restaurant menus in this project. As OCR engine, I've used Google Cloud Platform's Vision API. In order to use this API, I received an Application Key from Google and used it in this application. Sent menu image to Google servers using the REST API.

Lucene is an information retrieval library implemented with Java. It allows to perform complex queries and get results quickly over the big dataset. In this project I associated recipes with the meals.

A meal is modeled as an object which has ingredient fields mapped by its recipe. A meal and its ingredients constitute a document. Lucene uses inverted index structure to store these documents.

Each meal and ingredient has a list of documents that are related with them. By this way, by looking at the document list of a particular meal or ingredient, I can see all the occurrences of specific meal or ingredient. This structure allows to get results quickly from complex queries which include many "and", "or", "not" operations.

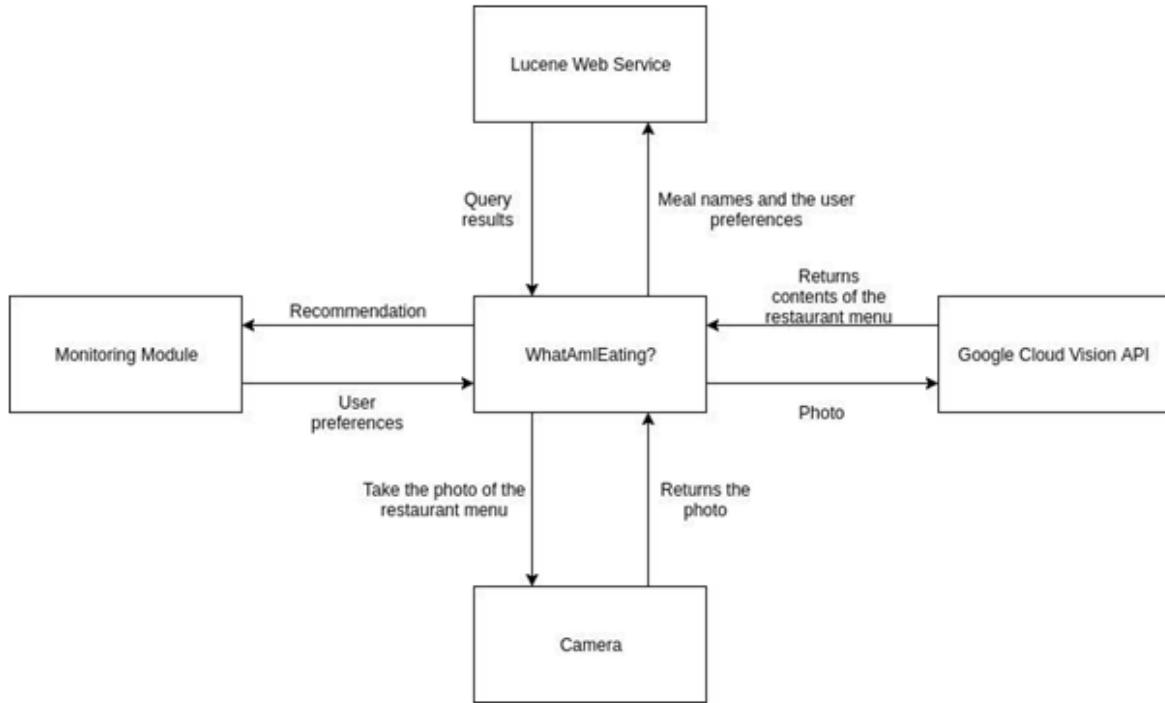
4 -) Method

First, I needed a dataset that includes meals and recipes to perform search for ingredients and meals. So, I've searched the recipes of the meals from the Internet. Parsed ingredients from the recipes and sent them to Lucene Library. I created documents and indexed them by using Lucene Library. By this way, constructed a dataset which is ready to perform search to get various information from it.

After that, analyzed images of the restaurant menu to get meals' names. I've used Google Cloud Vision API for this process. After getting contents of the restaurant menu, I sent the contents along with user preferences to Lucene web service implemented by me. Lucene web service creates queries according to incoming data and sends results back to the user.

Since there are many different meals in restaurant menus, I expanded the dataset periodically and started index process over and over again. The dataset still needs to be expanded.

5 -) Technical Design



There are two main components of this application. The first component of the application; converts image of the restaurant menu to text in order to get meal names. The second component of the application creates dataset which consists of the meals and their recipes and performs search over this dataset to give recommendation to the user.

The application communicates with two servers throughout the recommendation process. The first server is responsible for converting image to text. The application sends the text information to the second server in order to get recommendation. The second server is responsible for creating queries according to restaurant menu and user preferences. After that the server sends results to the application. This application will run on mobile android devices.

User takes the photograph of the restaurant menu. The application sends this photo to Google Cloud Vision API and requests contents of the menu. After getting contents of the restaurant menu, the application performs validation over incoming text information to eliminate the price and extra information about meals. The application sends meal names and the user preferences to Lucene server and requests recommendation. Lucene server creates queries according to incoming data and returns the results to the application. The application renders incoming data and constructs user interface to show the results to the users.

Some important points about the application; since the OCR Engine cannot read the texts on the photo with 100% accuracy, I need an algorithm to simulate the text I have obtained from the dataset. For this purpose I've used the *Jaro-Winkler String Similarity Algorithm*.

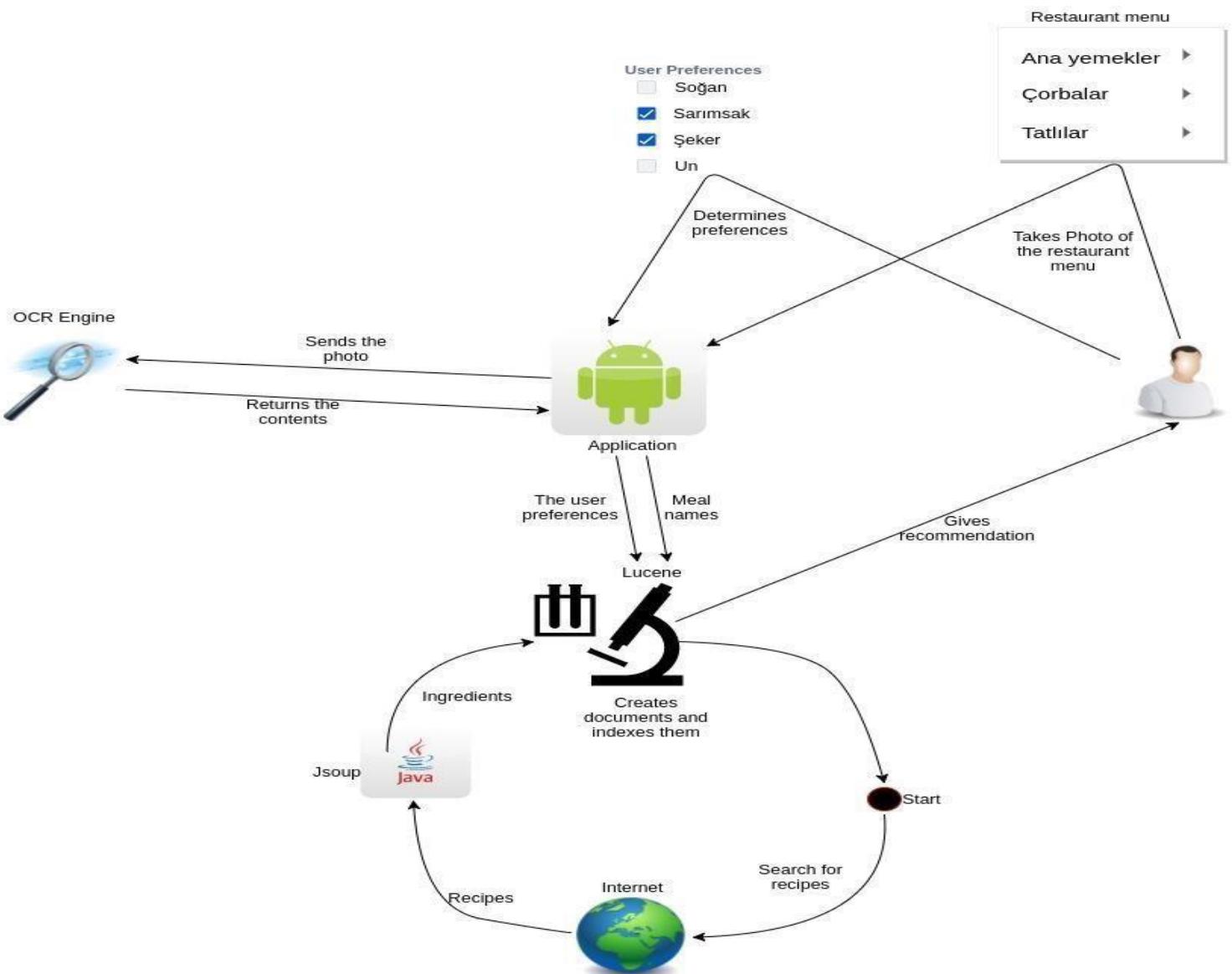
The Jaro distance d_j of two given strings s_1 and s_2 is

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{otherwise} \end{cases}$$

Where:

- m is the number of matching characters
- t is half the number of transpositions

Overview of the project;



I've taken recipes with software scripts from various websites. However, these recipes contained rather unnecessary words. They had to be cleaned. For this purpose stopwords were created and used.

-----> 700 gram dana kıyma	-----> 2 dilim büyük boy fileto levrek balığı
dana kıyma	levrek
-----> 300 gram kuzu kıyma	-----> 1/2 adet küçük boy kırmızı soğan
kuzu kıyma	soğan
-----> 1 diş sarımsak	-----> 4 adet cherry domates
sarımsak	domates
-----> 2 adet kuru soğan	-----> 1 yemek kaşığı kapari turşusu
soğan	kapari turşusu
-----> 1 çay kaşığı pul biber	-----> 2 yemek kaşığı dilimlenmiş siyah zeytin
pul biber	siyah zeytin
-----> 1 çay kaşığı karabiber	-----> 2 yemek kaşığı zeytinyağı
karabiber	zeytinyağı
-----> 1 yemek kaşığı acı biber salçası	-----> 1 dal taze biberiye
biber salçası	biberiye

An example of finding meals according to ingredient preferences.

“Nane” is wanted by user; “Tavuk”, “Sarımsak” and “Soğan” are not wanted by user.

```
*:* +ingredient:nane -ingredient:tavuk -ingredient:sarımsak -ingredient:soğan
Total Results :: 4
Meal : yogurt-corbasi, Score : 4.044036 id: 5
et,un,yumurta,yoğurt,tuz,tereyağı,biber,nane

-----
Meal : yayla-corbasi, Score : 3.8255622 id: 2
pirinç,tereyağı,nane,pul biber,tuz,yoğurt,un,yumurta,limon

-----
Meal : dugun-corbasi, Score : 3.8255622 id: 18
kuzu eti,un,yoğurt,yumurta,limon,tuz,tereyağı,biber,nane

-----
Meal : tarhana-corbasi, Score : 3.7276778 id: 4
tarhana,et,ayçiçek yağı,domates salçası,tuz,karabiber,pul biber,nane
```

Various image processing techniques (noise removing, grayscaling, resizing, deskewing and border removal) were applied to the photos in order to recognize Turkish characters.

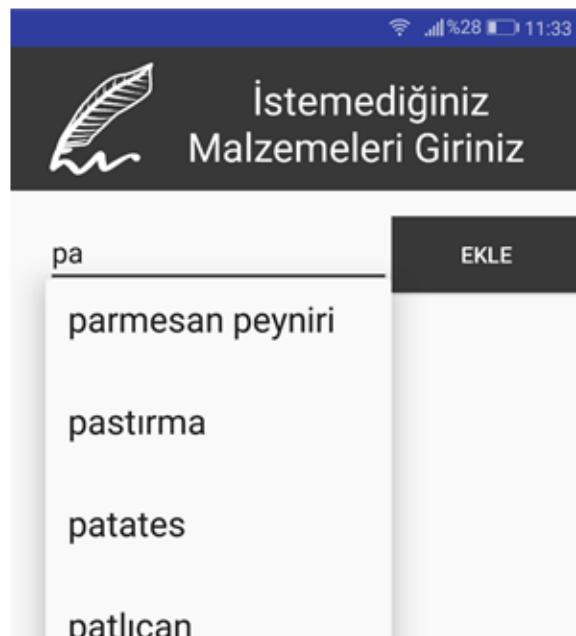
HER GÜNE ÖZEL SULU YEMEK ÇEŞİTLERİ		ÇORBA ÇEŞİTLERİ		HER GÜNE ÖZEL ÇORBA ÇEŞİTLERİ SULU YEMEK ÇEŞİTLERİ	
Sebzeli Kebap	6.50₺	İşkembe Çorbası	5.00₺	Sebzeli Kebap	İşkembe Çorbası
Orman Kebabı	6.50₺	Kellepaça	6.00₺	Orman Kebabı	Kellepaça
Biber Dolma	6.50₺	Ezogelin Çorbası	3.00₺	Biber Dolma	Ezogelin Çorbası
Taze Fasulye	6.00₺	Mercimek Çorbası	3.00₺	Taze Fasulye	Mercimek Çorbası
Yaz Türüsü	6.00₺	Tavuk Suyu Çorbası	3.00₺	Yaz Türüsü	Tavuk Suyu Çorbası
İzmir Köfte	6.50₺	PİLAV ÇEŞİTLERİ		İzmir Köfte	İzmir Köfte
Fırın Köfte	6.50₺	Pirinç Pilavi	3.00₺	Fırın Köfte	Fırın Köfte
Sebzeli Köfte	6.50₺	Bulgur Pilavi	3.00₺	Sebzeli Köfte	Pirinç Pilavi
İslim Köfte	6.50₺	Arnavut Ciğeri	7.00₺	İslim Köfte	Bulgur Pilavi
Hasanpaşa Köfte	6.50₺	Adınbudo Köfte	6.50₺	Hasanpaşa Köfte	Arnavut Ciğeri
Ekşili Köfte	6.50₺	Izgara Köfte	7.50₺	Ekşili Köfte	Adınbudo Köfte
Kuru Fasulye	5.00₺			Kuru Fasulye	Izgara Köfte

6 -) Application

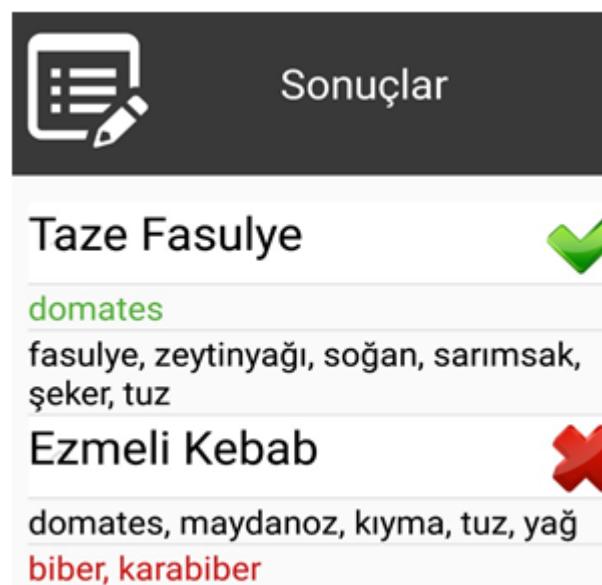
Home screen;



While the user reporting nutrition preferences; in terms of convenience to the user and to see the materials in the system, application has the autocomplete feature.



While the user seeing the results, the color of materials ,that user wants to use, is green; the color of materials, which user is not interested in, is black; and the color of materials, which user does not want to use, is red.



Results;



Results;

Sonuçlar
Saray Katmer
Şiş Kebap
Domatesli Kebap
Taskebap
Et Sote
Elbasan Tava
Ali Nazik
Arnavut Cigeri

TAMAM

Restaurant menu;

• TASKEBAP	9.00	TL
• DANA TUZLAMA	9.00	TL
• FIRIN KEBAP	7.50	TL
• ORMAN KEBAP	7.50	TL
• DOMATES KEBAP	7.50	TL
• SEZZELİ KEBAP	7.50	TL
• PATLICAN KEBAP	7.50	TL
• ET SOTE	7.50	TL
• ÇOBAN KAVURMA	8.00	TL
• SAÇ KAVURMA	7.50	TL
• ELBASAN TAVA	6.00	TL
• BOSTAN KEBAP	7.00	TL
• HÜNKAR BEĞENDİ	7.00	TL
• İSLİM KEBAP	7.00	TL
• BEZELEYELİ KEBAP	6.00	TL
• ET YAHNI	7.00	TL
• ALİ NAZIK	7.00	TL
• ETLİ GÜVEC	7.00	TL
• ET HAŞLAMA	8.00	TL
• KANARYA KEBAP	7.00	TL
• SARAY KATMER	7.00	TL
• ARNAVUT CİĞERİ	6.50	TL
• KAŞARLI CİĞER	7.00	TL

Details;

Sonuçlar
Saray Katmer
Şiş Kebap
Domatesli Kebap
Taskebap
Et Sote

TAMAM

Unwanted Ingredients Screen:



Restaurant Menu Photograph:

• TASKEBAP	9.00	TL
• DANA TUZLAMA	9.00	TL
• FIRIN KEBAP	7.50	TL
• ORMAN KEBAP	7.50	TL
• DOMATES KEBAP	7.50	TL
• SEBZELİ KEBAP	7.50	TL
• PATLICAN KEBAP	7.50	TL
• ET SOTE	7.50	TL
• ÇOBAN KAVURMA	8.00	TL
• SAÇ KAVURMA	7.50	TL
• ELBASAN TAVA	6.00	TL
• BOSTAN KEBAP	7.00	TL
• HÜNKAR BEĞENDİ	7.00	TL
• İSLİM KEBAP	7.00	TL
• BEZELYELİ KEBAP	6.00	TL
• ET YAHNİ	7.00	TL
• ALİ NAZIK	7.00	TL
• ETLİ GÜVEÇ	7.00	TL
• ET HAŞLAMA	8.00	TL
• KANARYA KEBAP	7.00	TL
• SARAY KATMER	7.00	TL
• ARNAVUT CİĞERİ	6.50	TL
• KAŞARLI CİĞER	7.00	TL

Results Screen:

Sonuçlar	
Saray Katmer	✓
Şiş Kebap	✓
Domatesli Kebap	✗
Taskebap	✗
Et Sote	✗
Elbasan Tava	✗
Ali Nazik	✗
Arnavut Cığası	✗

Details:

Sonuçlar	
Saray Katmer	✓
Şiş Kebap	✓
Domatesli Kebap	✗
Taskebap	✗
Et Sote	✗
Elbasan Tava	✗
Ali Nazik	✗
Arnavut Cığası	✗

7 -) Evaluation

When I review the results, I can evaluate the application as successful. In the taken menu photos; we can see that users can get suggestions with a good accuracy rate.

The most fundamental problem of the program may have trouble with is that the OCR results come with a low accuracy. In this case either no results will be found, or the Jaro-Winkler string similarity algorithm will resemble the incoming text to another meal. In this case, a meal that is not actually included in the photographed menu will be displayed as a recommendation to the user. For this reason, the program is more successful at the photographs taken on clear text menus.

On the other hand, the food dataset that I created is not enough. When we think about the variety of food and the naming of foods in different forms, application needs a very large dataset.

8 -) References

<https://cloud.google.com/vision/>

https://lucene.apache.org/core/7_0_1/core/overview-summary.html

[https://jsoup.org/apidocs/overview-summary.html/](https://jsoup.org/apidocs/overview-summary.html)

https://en.wikipedia.org/wiki/Optical_character_recognition

https://en.wikipedia.org/wiki/Jaro–Winkler_distance