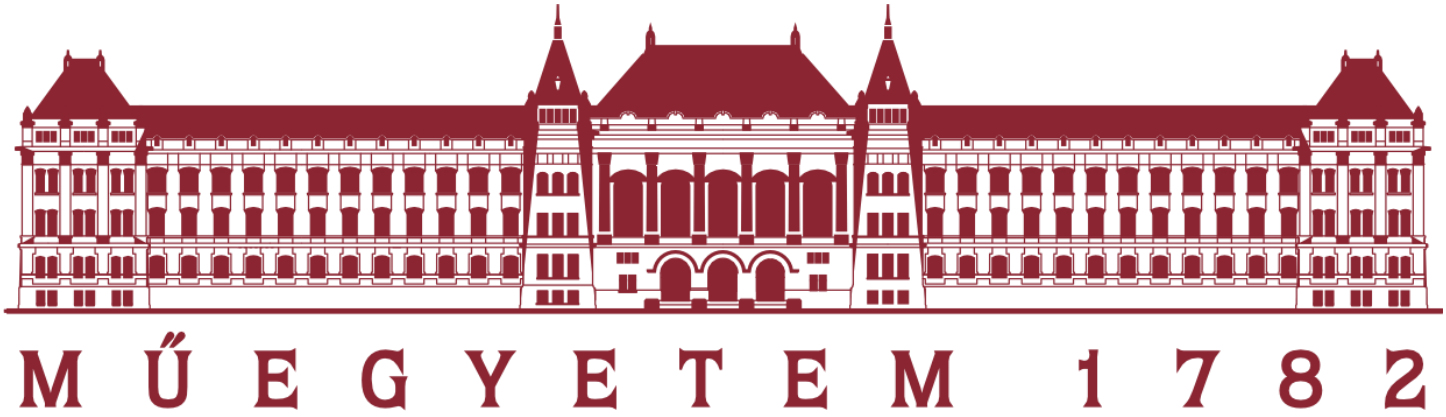
Student: **Akbar Hamidov**

Neptun: **FM8VEU**

Subject: **Client-side technologies**

Instructor: **Dr. Bence Kovari**

Task: **Face Recognition**



Face Recognition Application

1. Introduction

This application was developed mainly in **JavaScript**. Main functionality of application is accepting URL of image of person and trying to estimate his/her characteristics. For the front-end design **Bootstrap** library was used and for the backend interaction **Angular** was used. For the API which is giving these estimation results for corresponding image **Microsoft Azure Cognitive System** API was used. The characters of person which were estimated by API are those: age; gender; smile (in % out of 100); glasses (has, has not); moustache (in % out of 100); beard (in % out of 100); happiness (in % out of 100); sadness (in % out of 100). For navigating back to main page there is navbar and home button in it. In the navbar, application name and about button was also added. Lastly, in the end of application (in every page) developer signature was added.

In the below, visualization of main page after successful API call showing estimation values. *(Figure 1.1)*

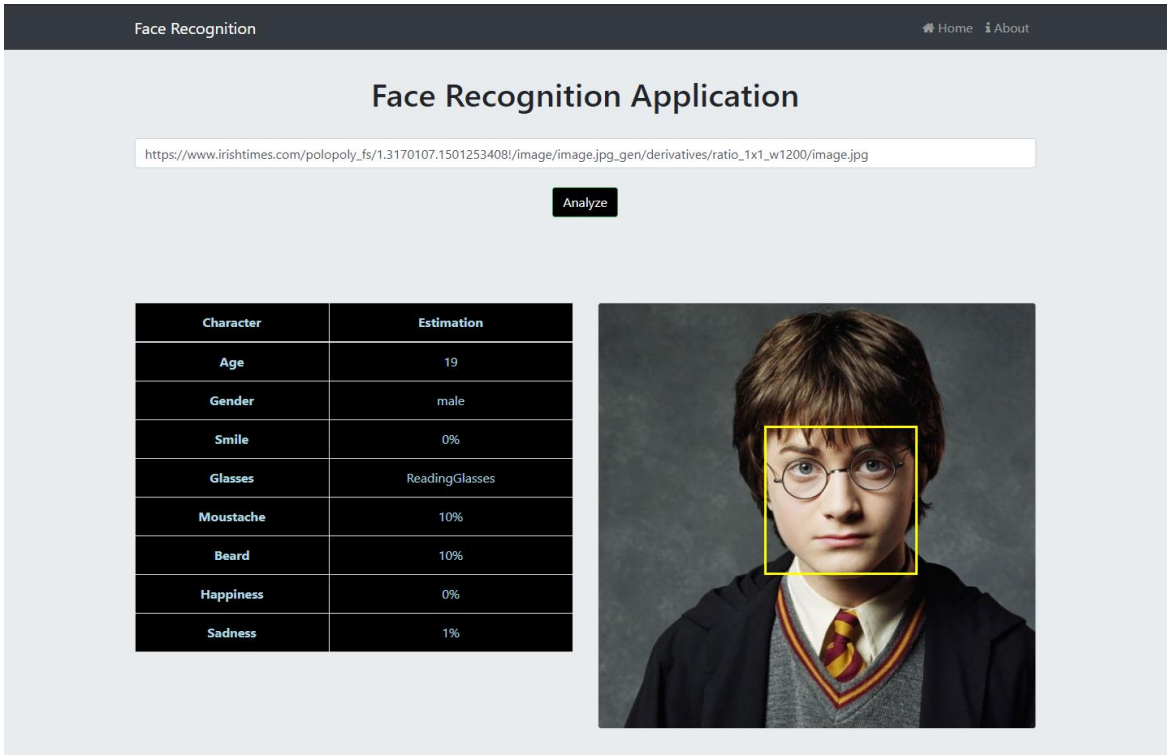


Figure 1.1

2. Architectural View

Angular is playing the role of heart in architectural model. Node modules were used building proper models for compiling application in ng server. Bootstrap components and code snippets from official documentation were used and modified in subsequent way due to developer's decision. For the API Microsoft Azure Cognitive Services was used and HTTPS protocol was used for calling API in safer mode. For managing application development life cycle and its versions GitHub was used. *(Figure 2.1)*

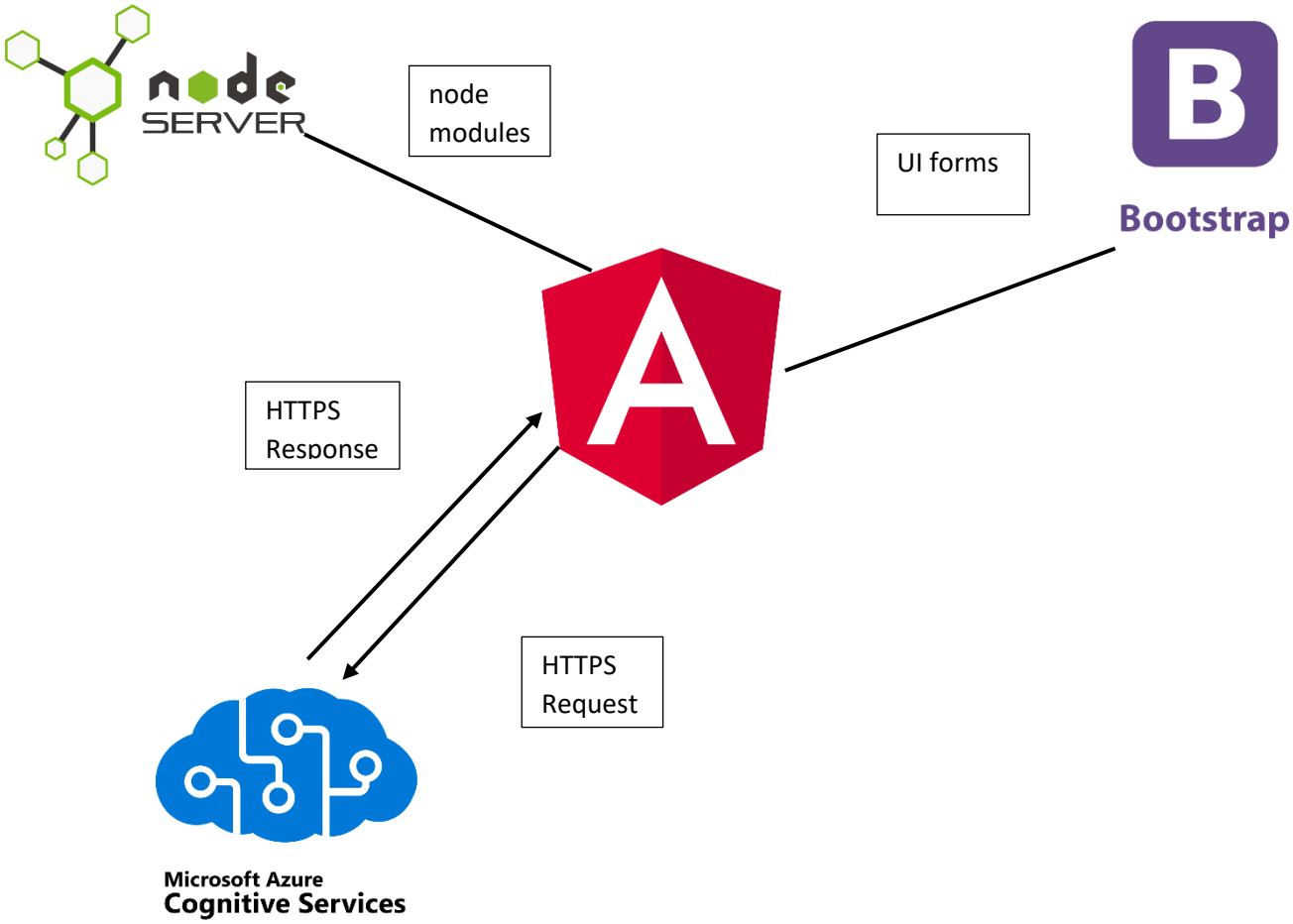


Figure 2.1

3. Source Code Snippets

- **Nav Menu Component** stores selector and calls corresponding .html and .scss files for navigation bar component.

src > app > nav-menu > TS nav-menu.component.ts > NavMenuComponent > ngOnInit

```
1   import { Component, OnInit } from '@angular/core';
2
3   @Component({
4     selector: 'app-nav-menu',
5     templateUrl: './nav-menu.component.html',
6     styleUrls: ['./nav-menu.component.scss']
7   })
8   export class NavMenuComponent implements OnInit {
9
10    constructor() { }
11
12    ngOnInit(): void {
13    }
14
15  }
16
```

Figure 3.1

- **About Component** stores selector and calls corresponding .html and .scss files for navigation bar component.

Figure 3.2

- **Face Analyze Component** stores main page selector and .html and .scss files.
  - Implements *onInit*.
  - *faceDetails (variable: any):* stores values of estimation which got from API call response.
  - *mainImg (variable: any):* stores image component which is inserted by user.
  - *getPersonDetails (function):* accepting URL of image as an input which entered by user in search box and called when analyze button pressed. Function finds corresponding values and store them into faceDetails variable.
  - *imageLoaded (function):* determines the location image after successful search result got.

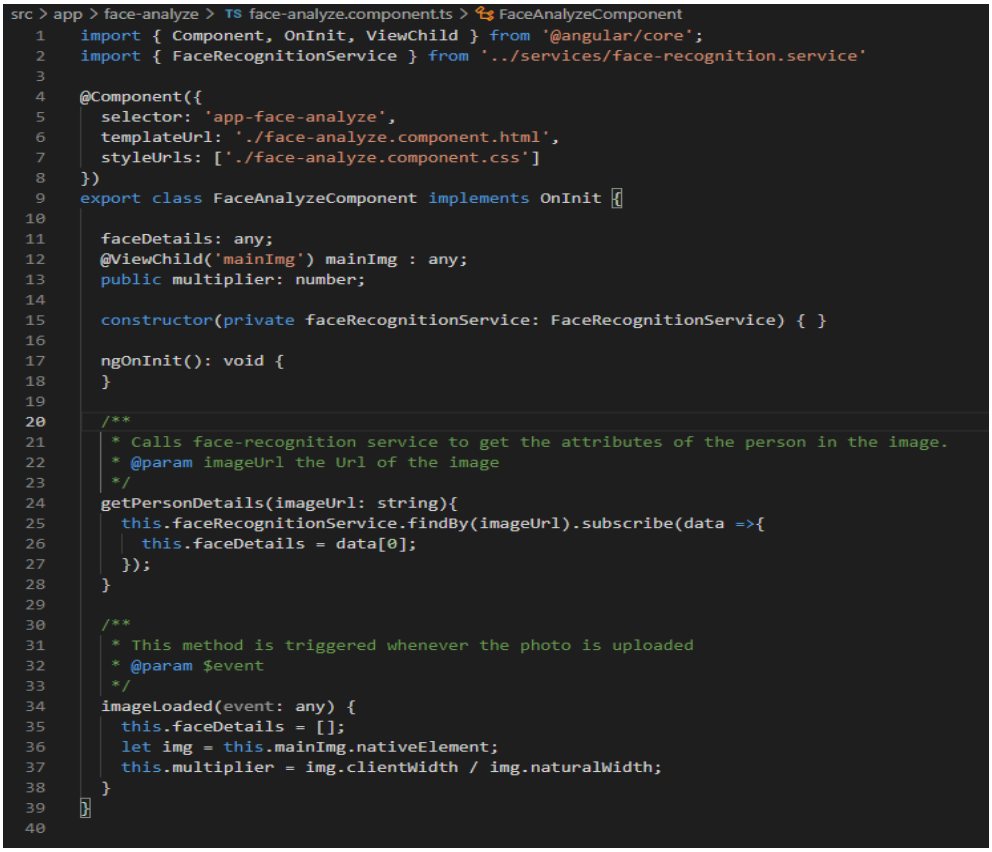
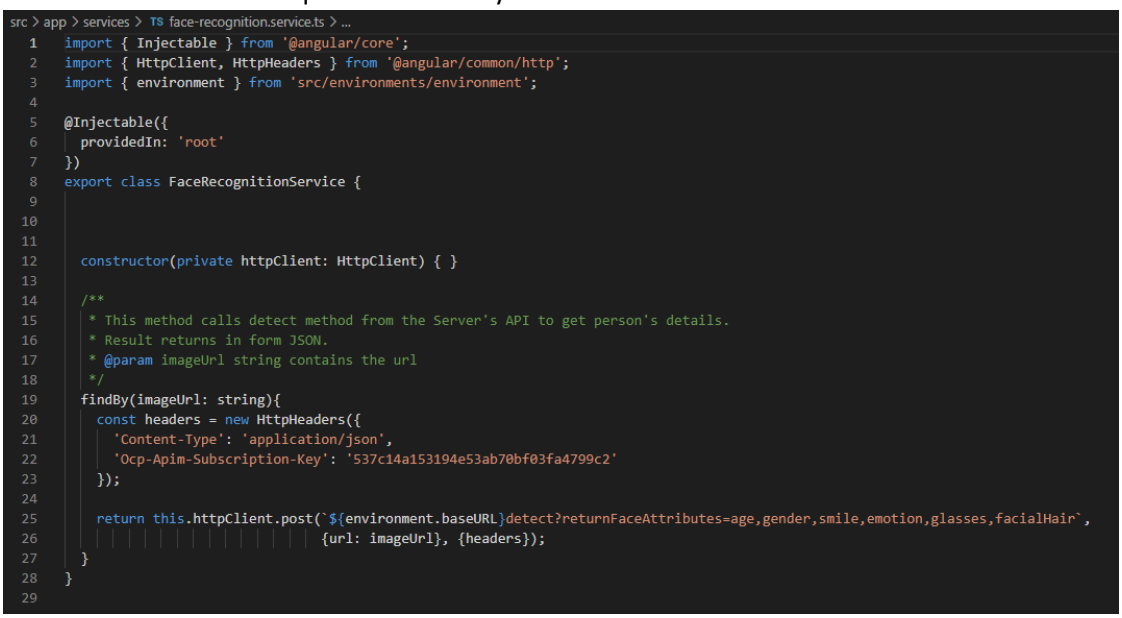


Figure 3.3

4. Client-server communication

- **Face Recognition Service** was created to connect API, and handling HTTPS requests and responses. HttpClient was passed with constructor and private keyword for used to data hiding. Function returns HTTPs response which HTTPS POST request was sent by HTTP client.



```
src > app > services > TS face-recognition.service.ts > ...
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { environment } from 'src/environments/environment';

@Injectable({
  providedIn: 'root'
})
export class FaceRecognitionService {



  constructor(private httpClient: HttpClient) { }

  /**
   * This method calls detect method from the Server's API to get person's details.
   * Result returns in form JSON.
   * @param imageUrl string contains the url
   */
  findBy(imageUrl: string){
    const headers = new HttpHeaders({
      'Content-Type': 'application/json',
      'Ocp-Apim-Subscription-Key': '537c14a153194e53ab70bf03fa4799c2'
    });

    return this.httpClient.post(`${environment.baseURL}detect?returnFaceAttributes=age,gender,smile,emotion,glasses,facialHair`,
                                {url: imageUrl}, {headers});
  }
}
```

Figure 3.4.1

- **Environment (typescript) file:** base URL of API Service and subscription key to service was declared in this file.

```
export const environment = {
  production: false,
  baseURL : 'https://faceapphw.cognitiveservices.azure.com/face/v1.0/'
};
```

Figure 3.4.1

5. Process sequences

Firstly, Angular loads index.html which loads **AppComponent.ts** by calling <app-root> selector. Every typescript component file loads template URL (.html file) and style URL (. scss, scss or others). In **app-component.component.html** fixed navigation bar and "router-outlet" selector is used to load **FaceRecognitionComponent.ts** and **AboutComponent.ts** according to the URL. Face recognition component send calls to FaceRecognitionService once image URL is given and "Analyze" button. **FaceRecongitionService.ts** send call to

https://faceapphw.cognitiveservices.azure.com/face/v1.0/detect?returnFaceAttributes=age,gender,smile,emotion,glasses,facialHair.

In addition, HTTP headers (content-type and API token) is also sent by request. Appropriate response gets received in FaceRecognitionComponent and displayed in html file. Result is displayed by mainly with Bootstrap.