



Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: LikedIt

Analysis Report

Zeynep Hande Arpakuş

Zeynep Ayça Çam

Muhammet Said Demir

Zeynep Nur Öztürk

Elif Beril Şaylı

Supervisor: Hamdi Dibeklioglu

Jury Members: Özcan Öztürk, Selim Aksoy

Contents

1. Introduction	4
2. Current System	4
2.1 OpenFace	5
2.2 Microsoft Azure Emotion API	5
3. Proposed System	5
3.1 Overview	5
3.2 Functional Requirements	6
3.3 Non-functional Requirements	7
3.3.1 Security	7
3.3.2 Usability	7
3.3.3 Cost	7
3.3.4 Performance	7
3.3.5 Extendibility	7
3.3.6 Marketability	8
3.4 Pseudo Requirements	8
3.5 System Models	9
3.5.1 Scenarios	9
3.5.2 Use-Case Model	16
3.5.3 Object and Class Model	17
3.5.4 Dynamic Models	19
3.5.4.1 Sequence Diagram	19
3.5.4.2 Sequence Diagram for Watching Process	20
3.5.4.3 Sequence Diagram for Getting Feedback as an Owner	21
3.5.4.4 Activity Diagram	22
3.5.5 User Interface	23
4. Other Analysis Elements	27
4.1 Consideration of Various Factors	27
4.1.1 Public Health	27
4.1.2 Public Safety	28
4.1.3 Public Welfare	28
4.1.4 Global Factors	28
4.1.5 Cultural Factors	28
4.1.6 Social Factors	28
4.1.7 Environmental Factors	28
4.1.8 Economic Factors	29

4.2 Risks and Alternatives	29
4.3 Project Plan	30
4.4 Ensuring Proper Team-Work	33
4.5 Ethics and Professional Responsibilities	33
4.6 New Knowledge and Learning Strategies	33
5. Glossary	34
6. References	35

1. Introduction

LikedIt is a website for watching videos. The difference between the other video platforms is the liking process. In LikedIt, a user can like a video by just watching it.

This program can understand that the user likes the video or doesn't by analyzing his face during the watching. While the user watches the video, LikedIt analyzes his face in real-time and determines the like rate for the video. Because, this process happens in real-time, the system doesn't record the user's face. After watching, the user can see this analysis as detailed feedback and even can share it with his friend on the desired platform if he wants.

Another difference between LikedIt and other video websites is the recommendation system. Because LikedIt has a more reliable liking process, this system can recommend videos to the user in a healthier way. Therefore, a more enjoyable watching experience is provided to the user.

The advantage of LikedIt, for the owners of videos, is detailed and reliable feedback about the video. The ones who upload a video to YouTube can see the like rate of viewers. Also, they can examine the parts viewers most interested in.

To sum up, LikedIt is a new way to watch and like videos. Because of this program, the phrase "Don't forget to click the like button." will be passed.

2. Current System

In the market, there are applications about Youtube videos. They understand whether a video is liked or disliked with sentimental analysis which is the automated process that uses AI to identify positive, negative and neutral opinions from the text. Sentiment analysis is widely used for getting insights from social media comments, survey responses, and product reviews, and making data-driven decisions [1]. There are also academic papers and implementations about predicting like-ratio on YouTube videos using sentiment analysis on comments. This research reveals that the results from this work showed that there is a weak correlation between the percentage of likes and the percentage of positive comments on a video [2]. On the other hand, to understand the value of video with comments, likes and dislikes of videos is problematic because it can be possible to manipulate like/dislike components with various programs and fake users. People's intention is ambiguous. Therefore, our application differs from that way with others. To manipulate likes and dislikes with fake faces and emotions is harder than manipulate these with fake user's comments and the rate of a like button.

Although there aren't any system which understands user's like and unlike about videos with investigating their faces, there is a various application which investigates facial behaviors and emotions. This section reveals the functionalities of main functions.

2.1 OpenFace

OpenFace is an open-source toolkit for facial analysis. OpenFace is a useful tool for computer vision, machine learning, and affective computing communities and will stimulate research in facial behavior analysis an understanding [3].

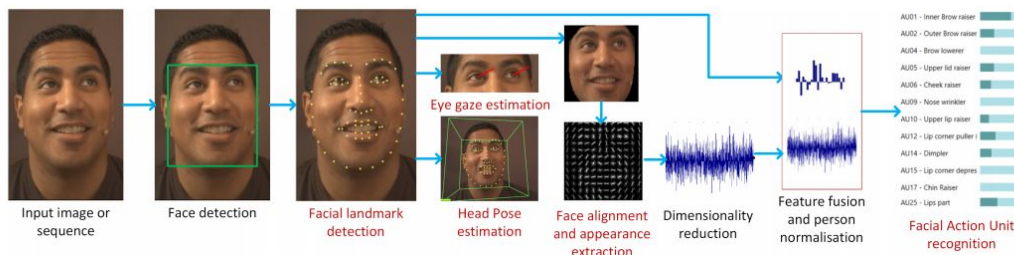


Figure 1: OpenFace [4]

2.2 Microsoft Azure Emotion API

Microsoft tool recognizes emotions which are anger, contempt, disgust, fear, happiness, neutral, sadness and surprise for each of detected faces. The artificial intelligence algorithms detects these emotions based on universal facial expressions, functioning even cross-culturally [5].

3. Proposed System

3.1 Overview

LikedIt is a website to watch videos. After users login the website with their Google account, the system becomes ready to show videos through YouTube. After signing in with a google account, the only thing the user should do is just clicking on a video.

The system pulls the youtube account information, such as videos that are uploaded by the user, from YouTube by using Google OAuth 2.0 API [6]. While the user watches the video, the system examines the face of the user and analyses the reactions by using deep learning technology. According to these

analyses, a feedback report is provided and sent to the owner of the video. However, in this process, the record of the user and his name is not sent to anyone. Moreover, the system doesn't record the video of the user. With cloud technology, the system can analyze the user's face in real-time without recording.

After watching, the user can see detailed feedback for himself. In this feedback, a gaze heatmap represents the parts the user look at while watching, a graphic shows the like rate for every 10 seconds of the video is shown. Also, the system recommends the scenes the user most react during the watching. After examining this feedbacks, the user can share these analyses with his friend on the desired platform.

Apart from these, the system recommends similar videos with the videos the user like. Also in the homepage, trend videos which are determined according to analyzes of all users are shown.

If there is any video uploaded by the user, the system shows him the feedback of the video. In this feedback, there is a graph shows for each scene, people's reactions. Also, in this feedback, the user can see the parts of the screen viewers looked at most during the watching.

3.2 Functional Requirements

- The user can create a profile.
- The user can input personal information.
- The program can suggest videos based on user data and previously accumulated data.
- The program can send notifications to the user regarding and events.
- The program can generate reports about the appreciation level of videos and send them to the owner of videos.
- The program can generate reports about which part of the videos gets the attention of the user and send them to the owner of videos.
- The program can show trend videos of the week.
- The user can share the link of videos and appreciation rate results of the videos with their friends.
- The user can select to share their personal data with developers to improve the project. By default, the user doesn't share their personal data.
- The owner of videos should be notified when the processing is done and research results are ready.

3.3 Non-functional Requirements

3.3.1 Security

- The application should not keep the data of the user's face.
- The application should provide security of personal data of the appreciation rate for videos. They should be private to the user.

3.3.2 Usability

- Our user interface will be easy to use for anyone who can use Youtube. The design of the site and mobile application should be easy to understand for them.
- The application should be available to everyone with regard to YouTube's age limitation for specific videos.
- The application should be stable in order to prevent any kind of interruptions.

3.3.3 Cost

- The application should be free of charge for all users.

3.3.4 Performance

- The application should analyze the face of the user in real-time.
- The appreciation rate for application should have high accuracy.

3.3.5 Extendibility

- This application can also be adapted to other platforms such as Netflix, Instagram, Ted Talks and any other platform on the internet that contains videos.

3.3.6 Marketability

- The application provides detailed feedback to the owner of the videos from the viewer's appreciation of data. Which includes an average appreciation rate for each second of video, the appreciation map that defines the average appreciation rate of the users from a specific country, age and gender groups. Therefore, the video uploaders will find it handy to use the LikedIt.
- Advertisement companies can measure how their advertisements affect end users.
- Also, companies can produce their own advertisement in light of this feedback without advertisement companies.

3.4 Pseudo Requirements

- LikedIt will be a web application.
- The system that will make the presentation analysis will be mainly written in Python language. However, some part of the project might be written in other languages.
- As a continuous integration tool, version control system, source code management we will use GitHub.
- If a customer requests a new feature for the product, that request will be charged extra, and it should be delivered on time without delaying the delivering the other required features.
- For real time video processing and neural network algorithms to implement and train models, external libraries and API's will be used.
- The language of the website will be Turkish and English.
- Licenses and Copyright of the used external libraries, API's, textbooks of the courses should be checked, and if necessary required permission will be taken from the copyright holders.
- Sensitive information (User image, e-mail, address) of the customers cannot be shared with 3rd party applications.
- The required feature needs to pass several tests before meeting with the end user.
- The software development cycle of the project will be in Object Oriented Programming manner.
- The system will use third-party libraries like openFace.

3.5 System Models

3.5.1 Scenarios

Scenario 1 - Sign In

Use Case: Sign In

Actors: User

Entry Condition(s): User launches the website of LikeIt and hits the “Sign In” button.

Exit Condition(s): After a successful login, the user is redirected to the main page.

Flow of Events:

1. On the Sign In page of LikedIt, the user clicks the “Sign In” button.
2. User redirected to Google account sign in page.
3. For a successful email/phone number and password pair, user is redirected to main page.

Alternative Flow of Events:

- A. Sign in operation fails:
 - a. User enters a wrong combination of email/phone number and password.
 - b. An error message to inform the user is displayed.
 - c. Sign in page of Google account comes again.

Scenario 2 - Watch Video

Use Case: Watch Video

Actors: User

Entry Condition(s): User signs in successfully.

Exit Condition(s): Video ends or user aborts it by going back to main page.

Flow of Events:

1. User clicks on a video which is not uploaded by them.
2. User is redirected to the video page.
3. Adjust camera scene opens.
4. With an internet connection, user watches the video until the end.

Alternative Flow of Events:

- A.: Video is aborted:
 - a. User clicks to “Return to Main Page” button while watching the video.
 - b. User is redirected to the main page.
- B. Internet Connection is lost:
 - a. A message is served to user about the internet connection absence and the video stops.

Scenario 3 - Adjust Face

Use Case: Adjust Face

Actors: User

Entry Condition(s): User clicks to a video to watch.

Exit Condition(s): User adjust their in order to system to understand.

Flow of Events:

1. View from the camera is shown to the user to see themselves.
2. User adjust their face until system recognizes a face.
3. Video starts.

Alternative Flow of Events:

- A. User cannot adjust their face:
 - a. User is asked to adjust their face again for 60 seconds.
 - b. Session ends and returns back to main page.

Scenario 4 - See Like Analysis

Use Case: See Like Analysis

Actors: User

Entry Condition(s): User watches a video successfully.

Exit Condition(s):

- User clicks to "Return to Analyzes" button.
- User clicks to "Return to Main Page" button.

Flow of Events:

1. User clicks to "See Analyzes" button after successfully watching the video.
2. User clicks on the "See Like Analysis" button.
3. User clicks to "Return to Analyzes" button.

Alternative Flow of Events:

- A. User returns to main page:
 - a. On the Like Analysis page, user clicks on "Return to Main Page" button.
 - b. User is redirected to main page.

Scenario 5 - See Heat Mapped Video

Use Case: See Heat Mapped Video

Actors: User

Entry Condition(s): User watches a video successfully.

Exit Condition(s):

- User clicks to “Return to Analyzes” button.
- User clicks to “Return to Main Page” button.

Flow of Events:

1. User clicks to “See Analyzes” button after successfully watching the video.
2. User clicks on the “See Heat Mapped Video” button.
3. User clicks to “Return to Analyzes” button.

Alternative Flow of Events:

- A. User returns to main page:
 - a. On the Heat Mapped Video page, user clicks on “Return to Main Page” button.
 - b. User is redirected to main page.

Scenario 6 - See Highlights

Use Case: See Highlights

Actors: User

Entry Condition(s): User watches a video successfully.

Exit Condition(s):

- User clicks to “Return to Analyzes” button.
- User clicks to “Return to Main Page” button.

Flow of Events:

1. User clicks to “See Analyzes” button after successfully watching the video.
2. User clicks on the “See Highlights” button.
3. User clicks to “Return to Analyzes” button.

Alternative Flow of Events:

- A. User returns to main page:
 - a. On the Highlights page, user clicks on “Return to Main Page” button.
 - b. User is redirected to main page.

Scenario 7 - Share Analysis Report

Use Case: Share Analysis Report

Actors: User

Entry Condition(s): User watches a video successfully.

Exit Condition(s): User shares the report.

Flow of Events:

1. User clicks to “See Analyzes” button after successfully watching the video.
2. User clicks on the “Share Analysis” button.
3. User clicks on the platform they want to share on from the list of platforms they can use to share the report.
4. User is redirected to analyzes page after sharing.

Alternative Flow of Events:

- A. User clicks outside of the list:
 - a. Platform list is successfully displayed to user.
 - b. User clicks somewhere outside of the list.
 - c. List disappears and analyzes page comes again.

Scenario 8 - See Average Like Analysis

Use Case: See Average Like Analysis

Actors: User

Entry Condition(s): User clicks on one of their own videos.

Exit Condition(s):

- User clicks to “Return to Feedbacks” button.
- User clicks to “Return to Main Page” button.

Flow of Events:

1. User clicks to “See Video Feedbacks” button in one of their own videos.
2. User clicks on the “See Average Like Analysis” button.
3. User clicks to “Return to Feedbacks” button.

Alternative Flow of Events:

- A. No available feedbacks:
 - a. An error message is displayed to the user, saying that no one watched their video yet.
- B. User returns to main page:
 - a. On the Average Like Analysis page, user clicks on “Return to Main Page” button.
 - b. User is redirected to main page.

Scenario 9 - See Average Heat Mapped Video

Use Case: See Average Heat Mapped Video

Actors: User

Entry Condition(s): User clicks on one of their own videos.

Exit Condition(s):

- User clicks to “Return to Feedbacks” button.
- User clicks to “Return to Main Page” button.

Flow of Events:

1. User clicks to “See Video Feedbacks” button in one of their own videos.
2. User clicks on the “See Average Heat Mapped Video” button.
3. User clicks to “Return to Feedbacks” button.

Alternative Flow of Events:

- A. No available feedbacks:
 - a. An error message is displayed to the user, saying that no one watched their video yet.
- B. User returns to main page:
 - a. On the Average Heat Mapped Video page, user clicks on “Return to Main Page” button.
 - b. User is redirected to main page.

Scenario 10 - See Average Highlights

Use Case: See Average Highlights

Actors: User

Entry Condition(s): User clicks on one of their own videos.

Exit Condition(s):

- User clicks to “Return to Feedbacks” button.
- User clicks to “Return to Main Page” button.

Flow of Events:

1. User clicks to “See Video Feedbacks” button in one of their own videos.
2. User clicks on the “See Average Highlights” button.
3. User clicks to “Return to Feedbacks” button.

Alternative Flow of Events:

- A. No available feedbacks:
 - a. An error message is displayed to the user, saying that no one watched their video yet.
- B. User returns to main page:
 - a. On the Average Heat Mapped Video page, user clicks on “Return to Main Page” button.
 - b. User is redirected to main page.

Scenario 11 - See Trending Videos

Use Case: See Trending Videos

Actors: User

Entry Condition(s): User successfully signed in and on main (newsfeed) page.

Exit Condition(s):

- User clicks to return to main page.
- User clicks to one of the videos on Trending Videos list.

Flow of Events:

1. On the buttons at the left edge of the page, user clicks on “See Trending Videos” button.
2. User clicks to “Return to Main Page” button and redirected to main page.

Alternative Flow of Events:

- A. User opens a video:
 - a. User views the Trending Videos list.
 - b. User clicks to one of the videos in the list and redirected to that video.

Scenario 12 - See Recommended Videos

Use Case: See Recommended Videos

Actors: User

Entry Condition(s): User successfully signed in and on main (newsfeed) page.

Exit Condition(s):

- User clicks to return to main page.
- User clicks to one of the videos on Recommended Videos list.

Flow of Events:

1. On the buttons at the left edge of the page, user clicks on “See Recommended Videos” button.
2. User clicks to “Return to Main Page” button and redirected to main page.

Alternative Flow of Events:

- A. User opens a video:
 - a. User views the Recommended Videos list.
 - b. User clicks to one of the videos in the list and redirected to that video.

Scenario 13 - See My Videos

Use Case: See My Videos

Actors: User

Entry Condition(s): User successfully signed in and on main (newsfeed) page.

Exit Condition(s):

- User clicks to return to main page.
- User clicks to one of the videos on My Videos list.

Flow of Events:

1. On the buttons at the left edge of the page, user clicks on “See My Videos” button.
2. User clicks to “Return to Main Page” button and redirected to main page.

Alternative Flow of Events:

- A. User opens a video:
 - a. User views the My Videos list.
 - b. User clicks to one of the videos in the list and redirected to that video.

Scenario 14 - Search Video

Use Case: Search Video

Actors: User

Entry Condition(s): User successfully signed in and on main (newsfeed) page.

Exit Condition(s):

- User clicks to return to main page.
- User clicks to one of the videos on Search Result page.

Flow of Events:

1. On the search bar at the top of the page, user types the the the required text and clicks on “Search” button or hits “Enter” key.
2. User clicks to “Return to Main Page” button and redirected to main page.

Alternative Flow of Events:

- A. User opens a video:
 - a. User views the Search Result page
 - b. User clicks to one of the videos in the page and redirected to that video.

3.5.2 Use-Case Model

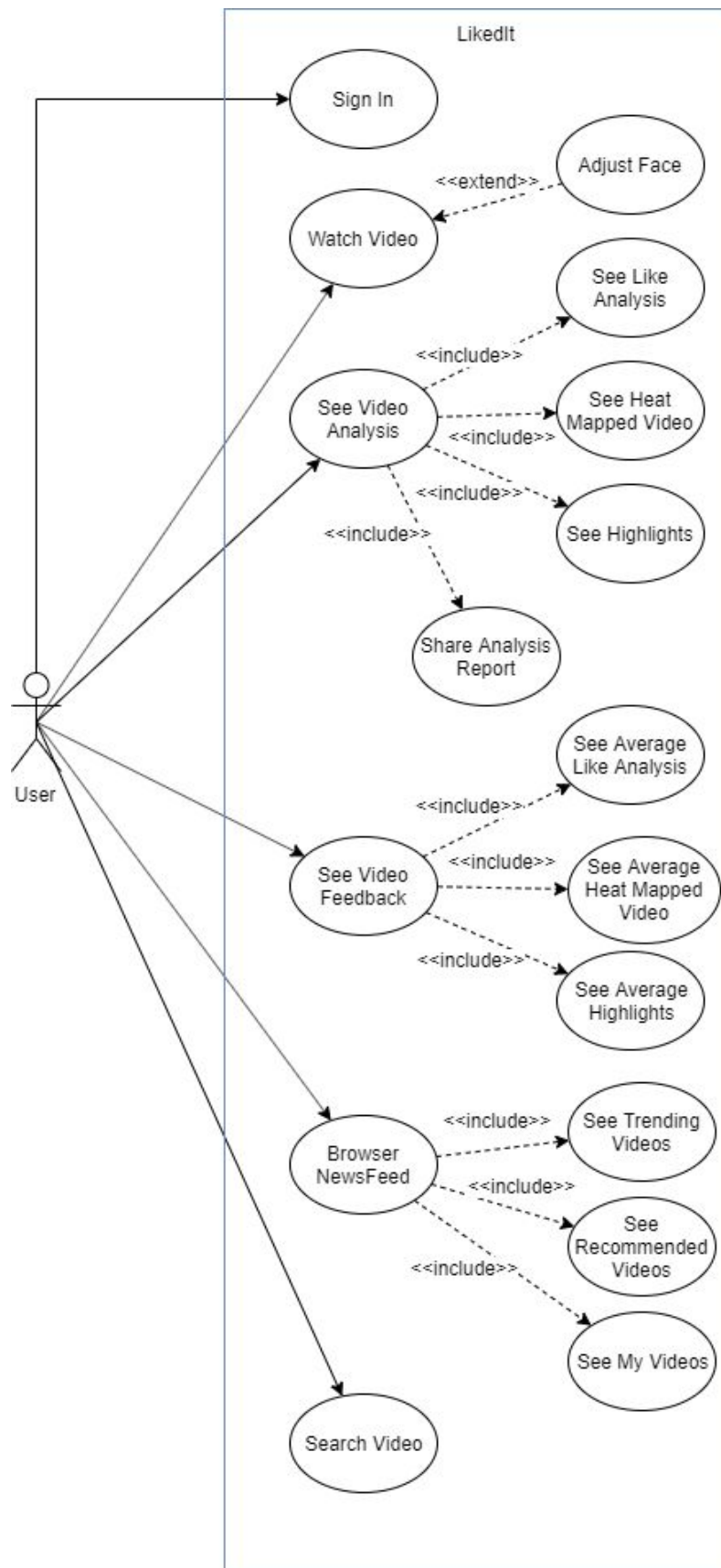


Figure 2: Use Case diagram

3.5.3 Object and Class Model

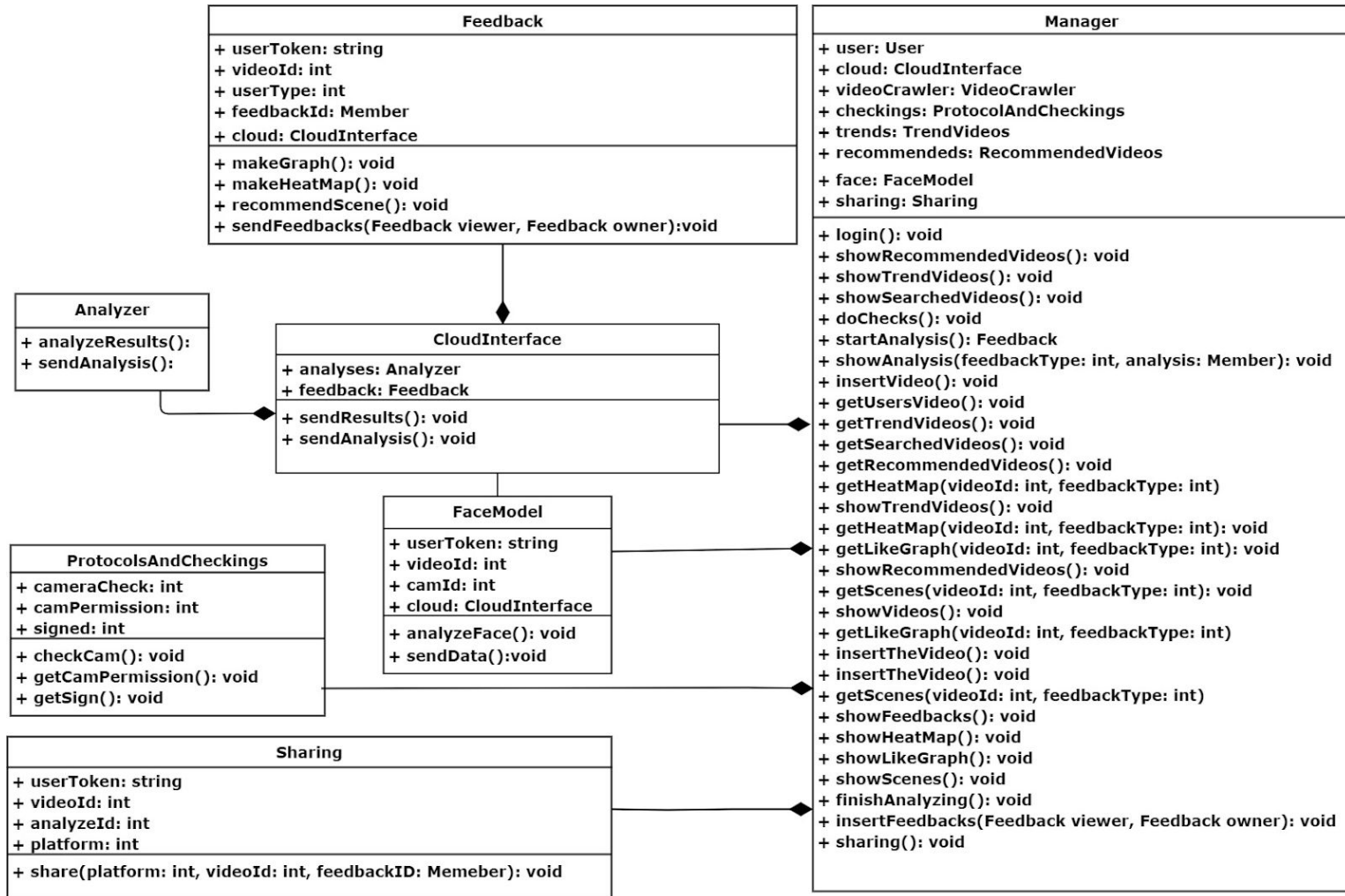


Figure 3: Class Diagram - 1

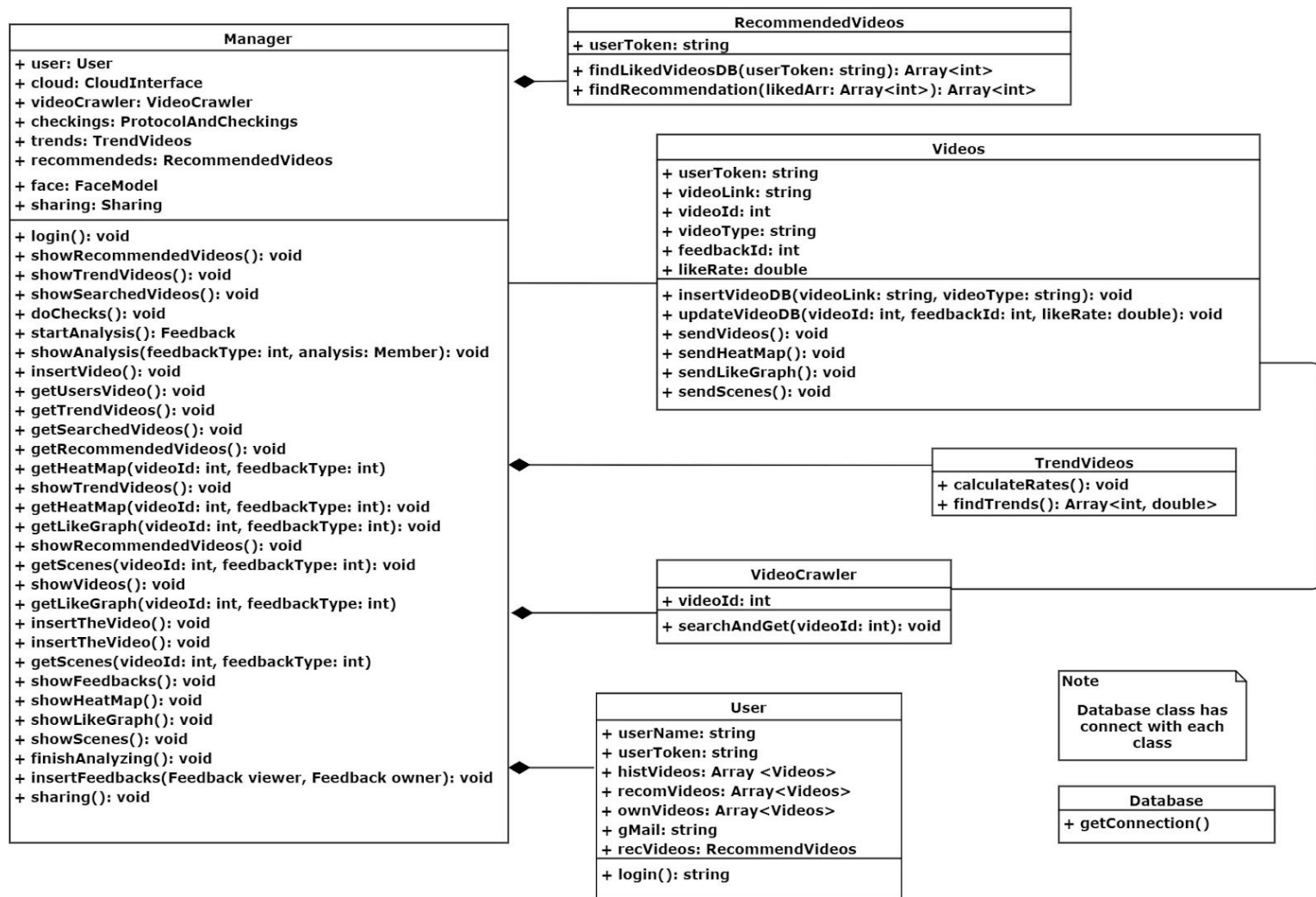


Figure 4: Class Diagram - 2

3.5.4 Dynamic Models

3.5.4.1 Sequence Diagram

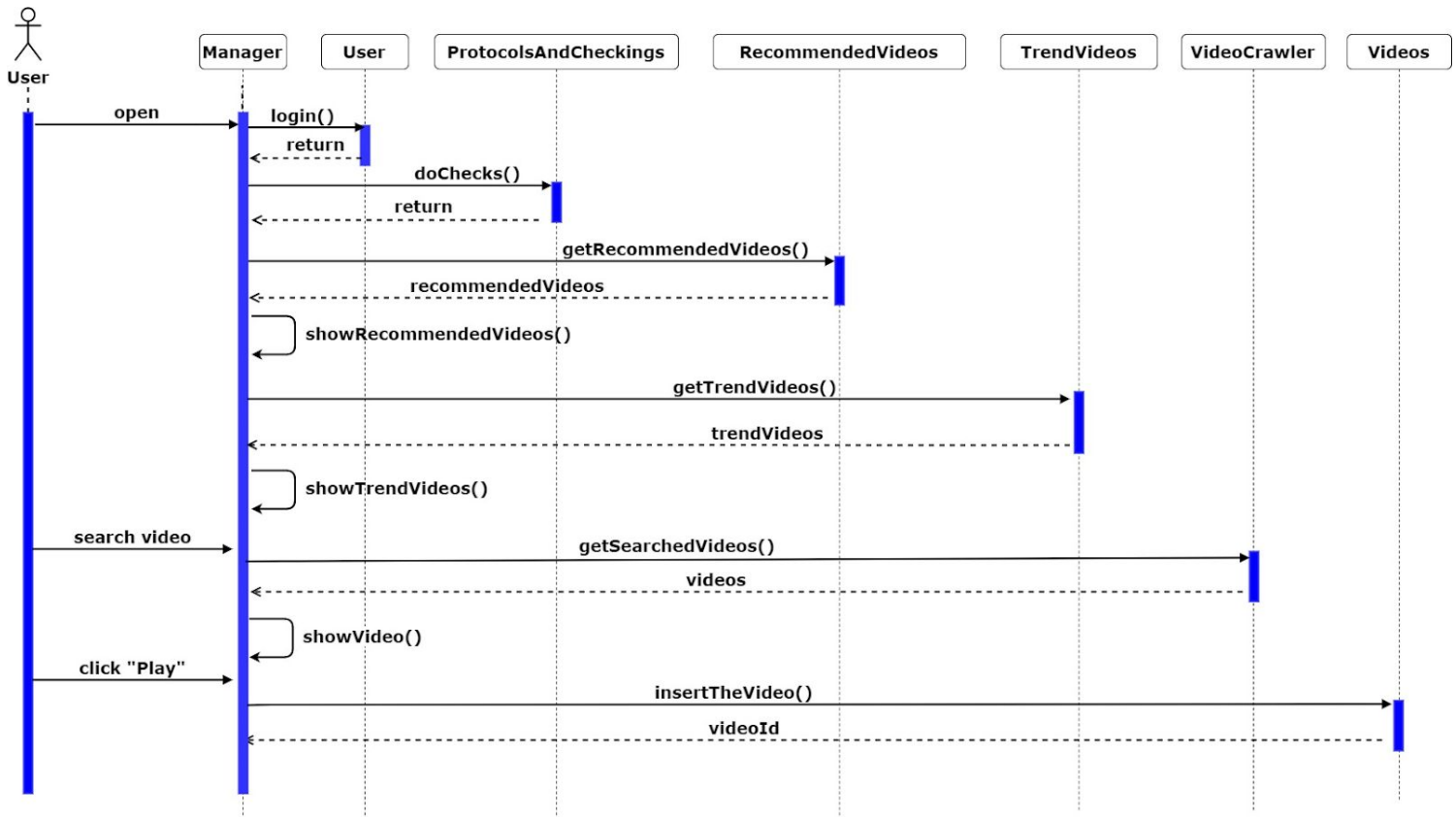


Figure 5: Sequence Diagram

This diagram represents the process until the watching phase. This process starts with signing in of the user. When the user signs in the system, the Manager class pulls the information from the user class. After this, the system does checkings of requirements for camera and permissions. Then, to prepare the home screen for the user, Manager class takes recommended videos for him and trend videos from RecommendedVideos and TrendVideos classes. After the user sees the homepage, if he searches a video, Manager class sends the words, the user wrote, to VideoCrawler class and this class finds the videos which fit these words through YouTube and return them to Manager class. When the user clicks one of these videos, the video is inserted to the database and id is created in Videos class.

3.5.4.2 Sequence Diagram for Watching Process

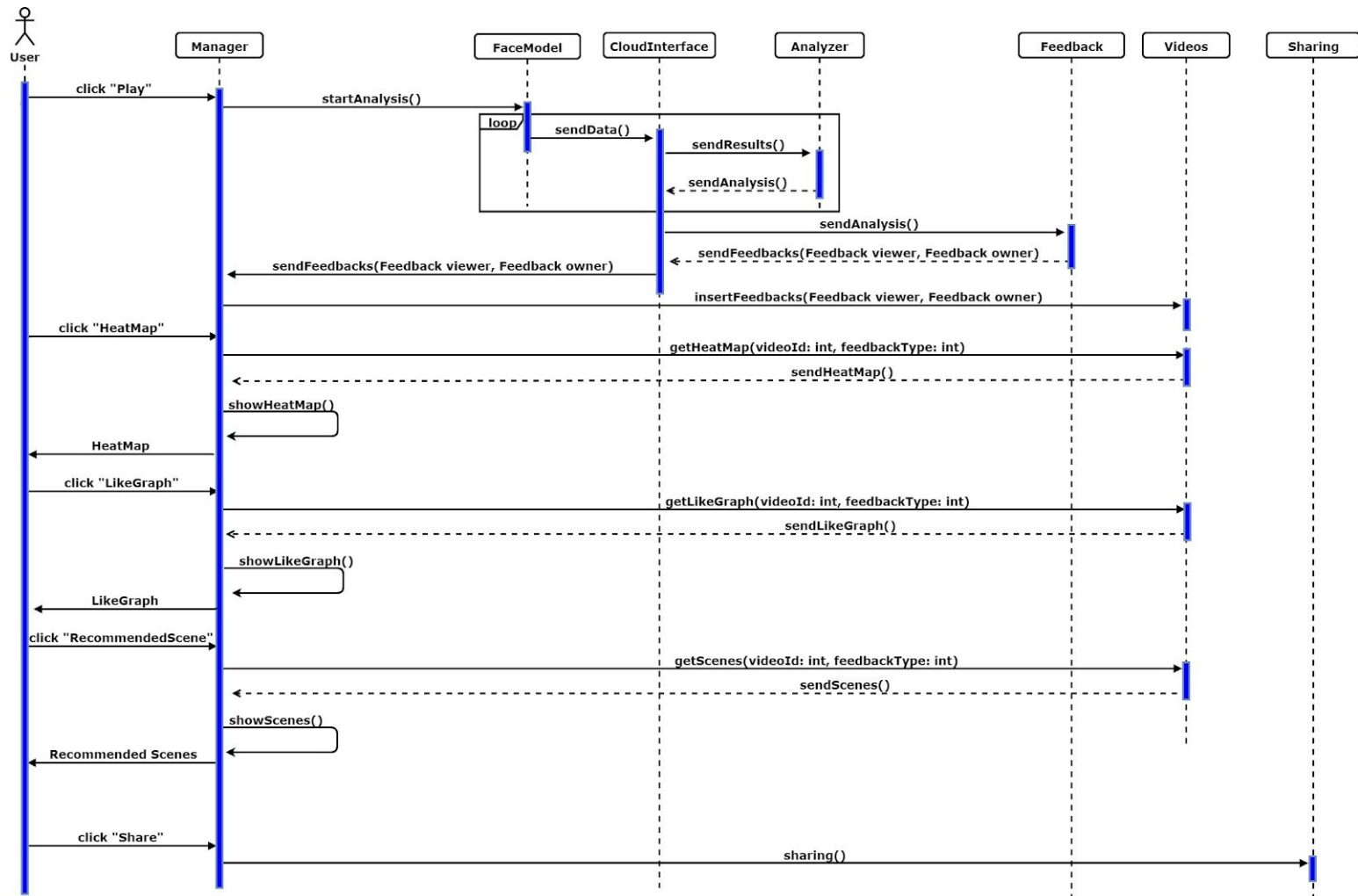


Figure 6: Sequence Diagram for watching process

This diagram shows the watching a video process of a user. The process starts with, the user clicking on a play button of any video. After the clicking, Manager class starts the analyze and calls FaceModel class. This class, during the watching, process the face of the user and sends the data about this processing to CloudInterface. This class sends the results coming from FaceModel to Analyzer class. Here, the results from CloudInterface are processed and analyzed. Then, these analyses are returned to CloudInterface. Then, CloudInterface sends the analyses which came from Analyzer and stored in this class, to Feedback class. This class converts the analyses into two types of feedbacks. One of these feedbacks is for the viewer of the video which is specific for the viewer, and the other one is for the owner of the video. Then, Feedback class sends this feedbacks to CloudInterface class and it sends them to Manager class. The feedback for the viewer of the video is shown in the screen, and meanwhile, two feedback is sent to Video class to inserted into the database. When the user sees the feedback screen, he can select a specific feedback such as a graphic represents the liking rate for each 10 seconds of the video or a heat

map represents the parts, the viewer most interested in, of the screen, or recommended scenes which took most reactions from the user during the watching. When the user clicks on one of them, Manager class pulls that feedback from Videos class and shows it on the screen. After all, if the user clicks on the 'Share' button, to share the analysis and link of the video in the desired platform, Manage class calls Sharing class.

3.5.4.3 Sequence Diagram for Getting Feedback as an Owner

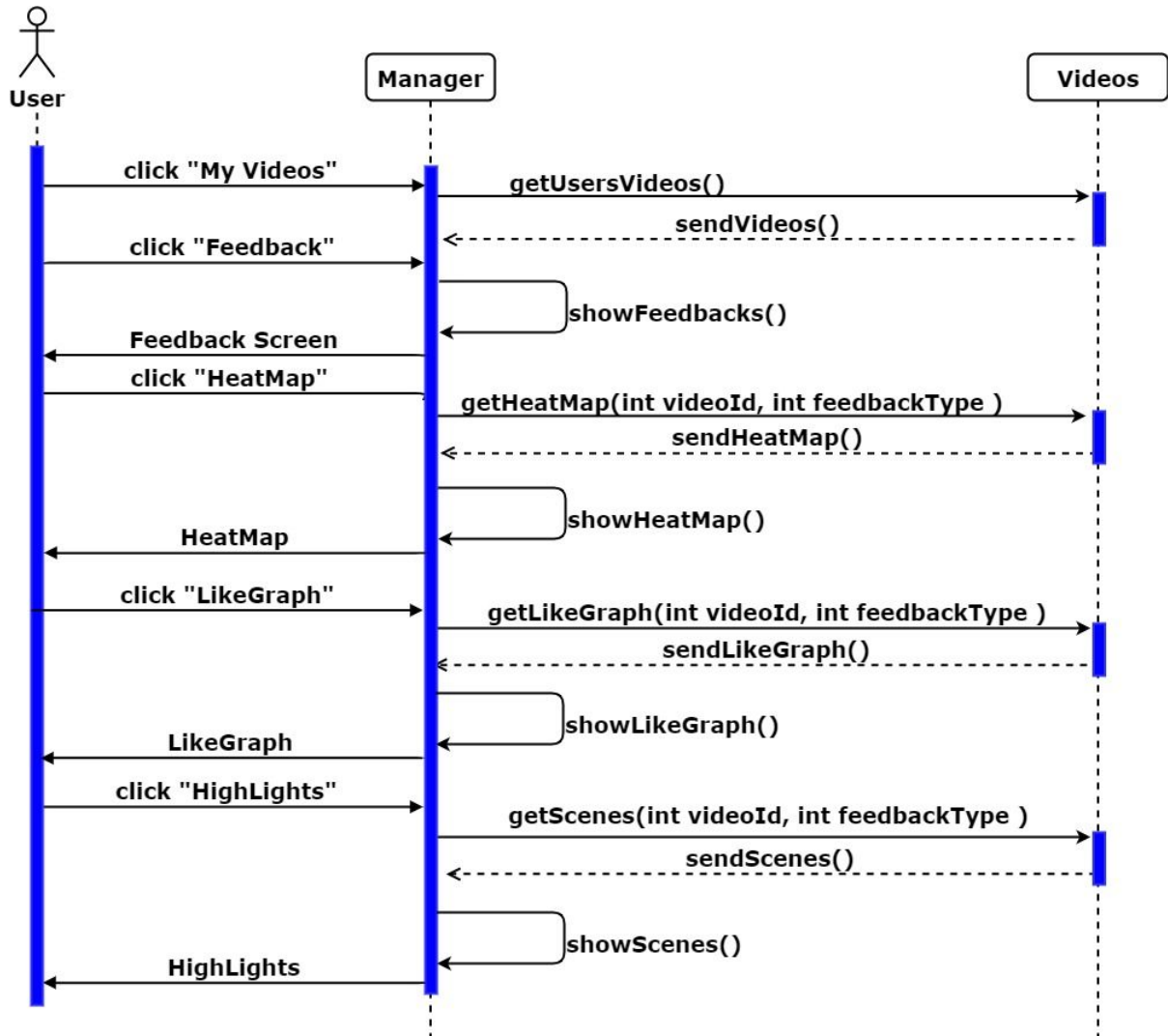


Figure 7: Sequence Diagram for getting feedback

This diagram shows the process of getting feedback for the owner of a video. This process starts with the user clicking on "My Videos" link in the homepage. For this screen, Manager class takes the videos of the user from Videos class and shows them on the screen. When the user clicks on the 'Feedback' button next to any video on the screen, the feedback screen is shown. On this screen, there are three types of feedback for a video. These are heat map, liking graph, and most recommended scenes. This feedbacks represent

the analyses of all viewers of the video. When the user selects one of them, Manager class takes this feedback from Videos class and shows it to the user.

3.5.4.4 Activity Diagram

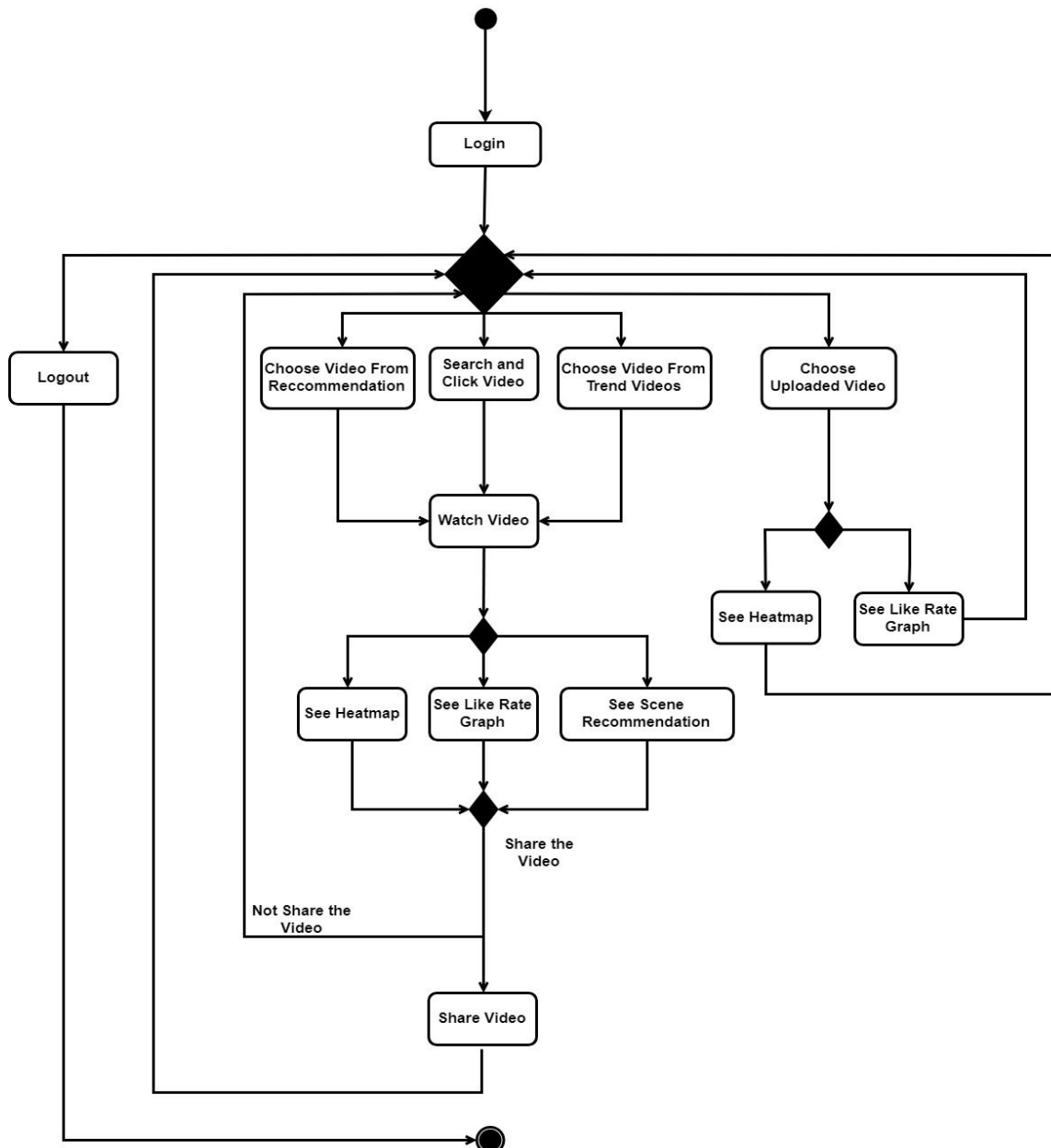


Figure 8: Activity Diagram

The program starts with the user signing in. First, there is the homepage on the screen. In this screen, the user can choose a video from recommendation videos or trend videos sections. Also, he can search for a new video. After choose a video and watch it, the user can see three types feedback which are a heat map, represents the parts the user look at during the watching, of the screen and a liking rate graph which shows liking rates of the user for each 10 seconds of the video and recommended scenes which are the user give reaction most in

the video. After examining this feedbacks, the user can share this video's link and his analyses with his friends on the desired platform. Also, if the user is an owner of any video, he can see these videos feedbacks too. This feedbacks don't include any personal information about the viewers. In this feedbacks, the owner can see a heat map represents the parts viewers looked at most, generally. Also, there is a like rate graph shows the average like rates of the viewers for each 10 of the video. The owner can see the most recommended scenes too.

3.5.5 User Interface

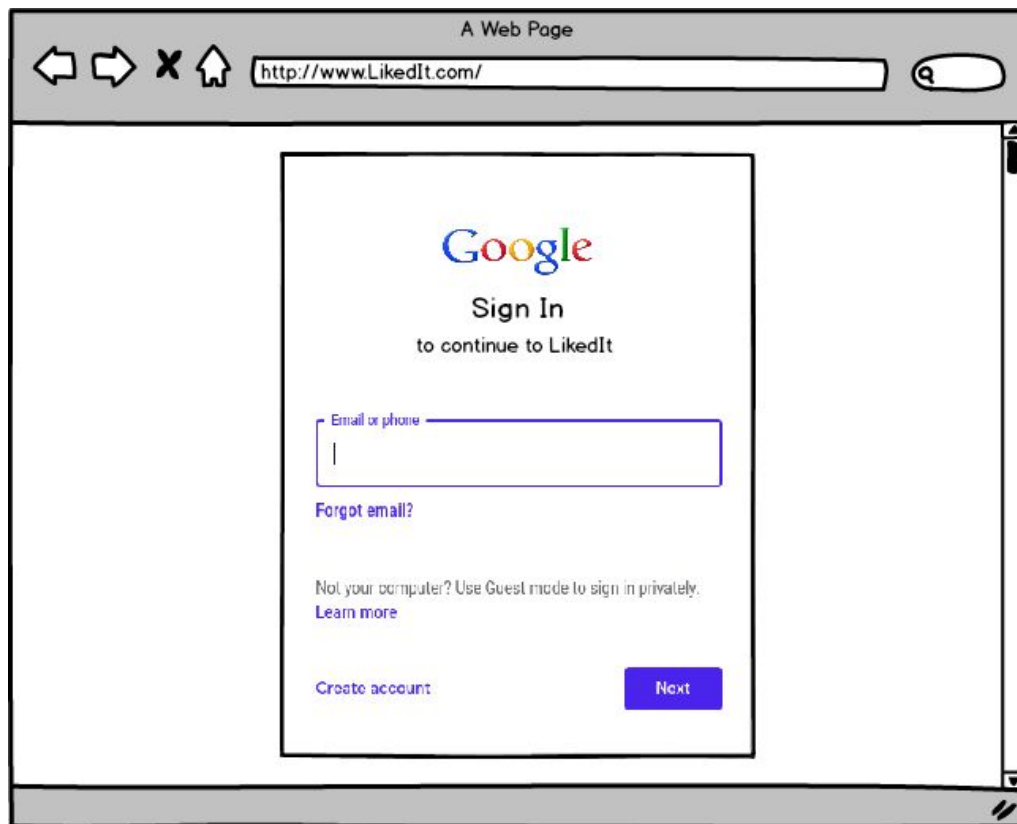


Figure 9: Sign in screen

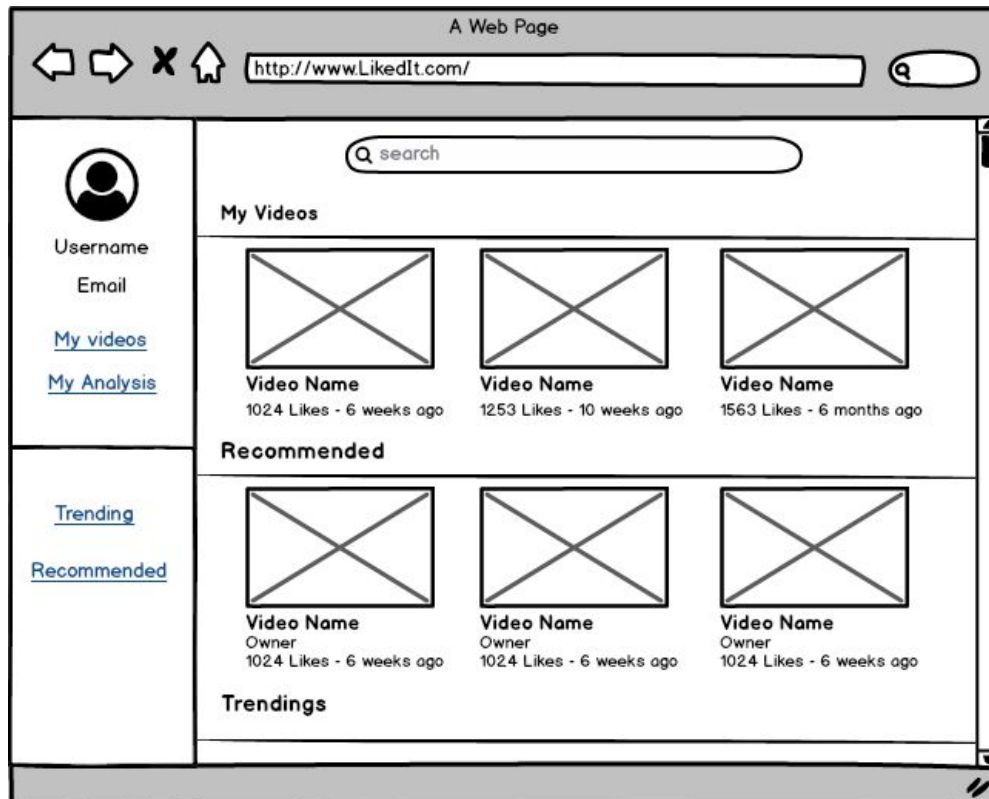


Figure 10: Homepage screen

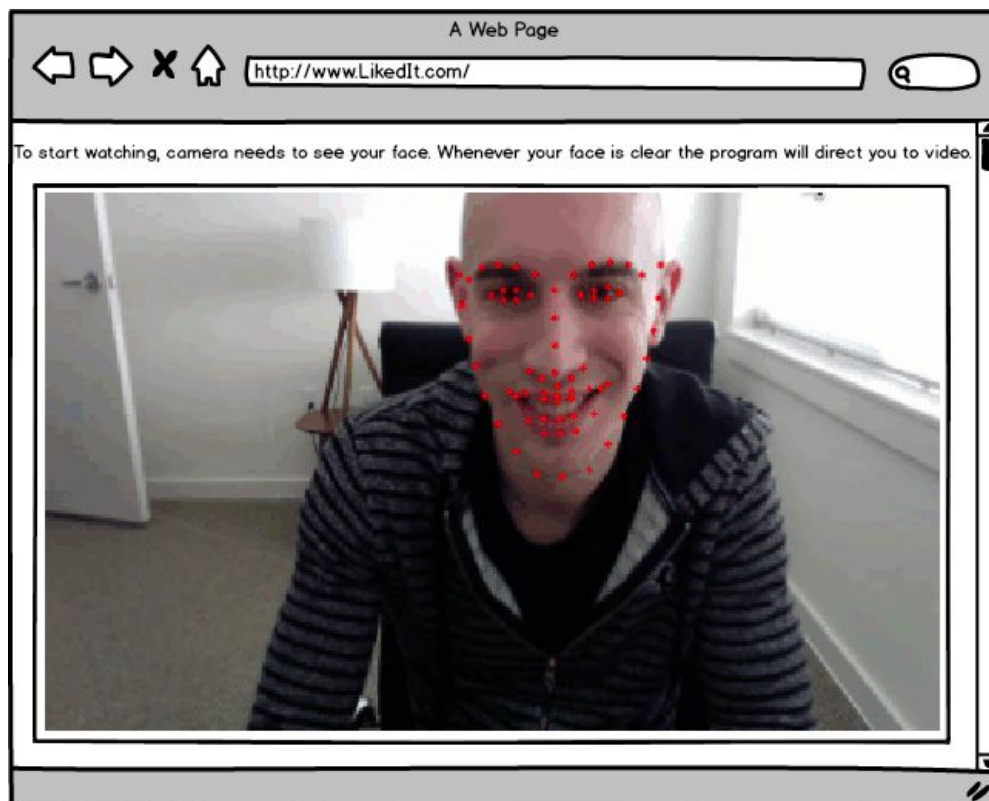


Figure 11: Face detecting



Figure 12: After watching, feedback screen

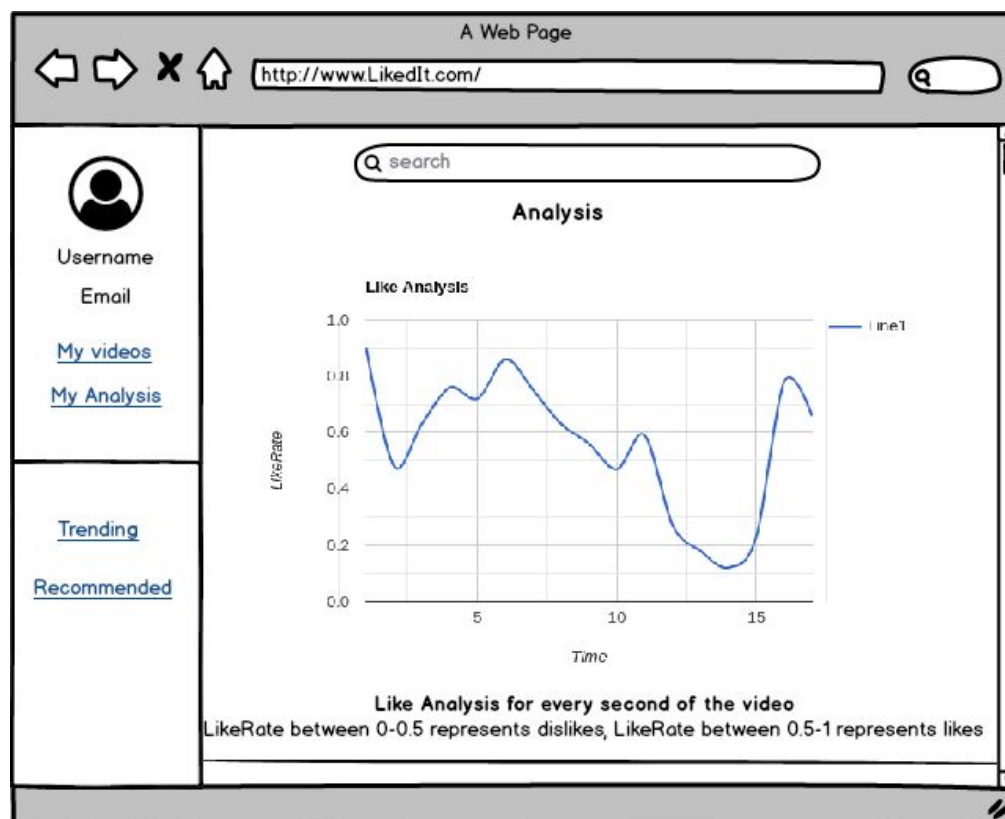


Figure 13: Like rate graph

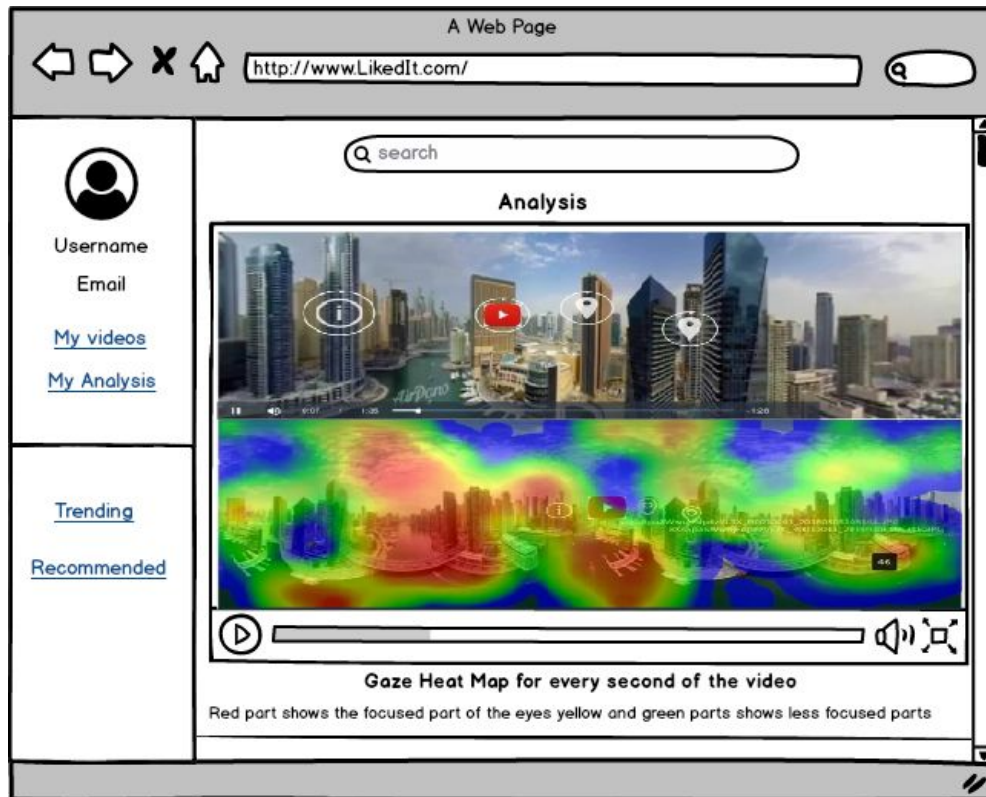


Figure 14: Heat map

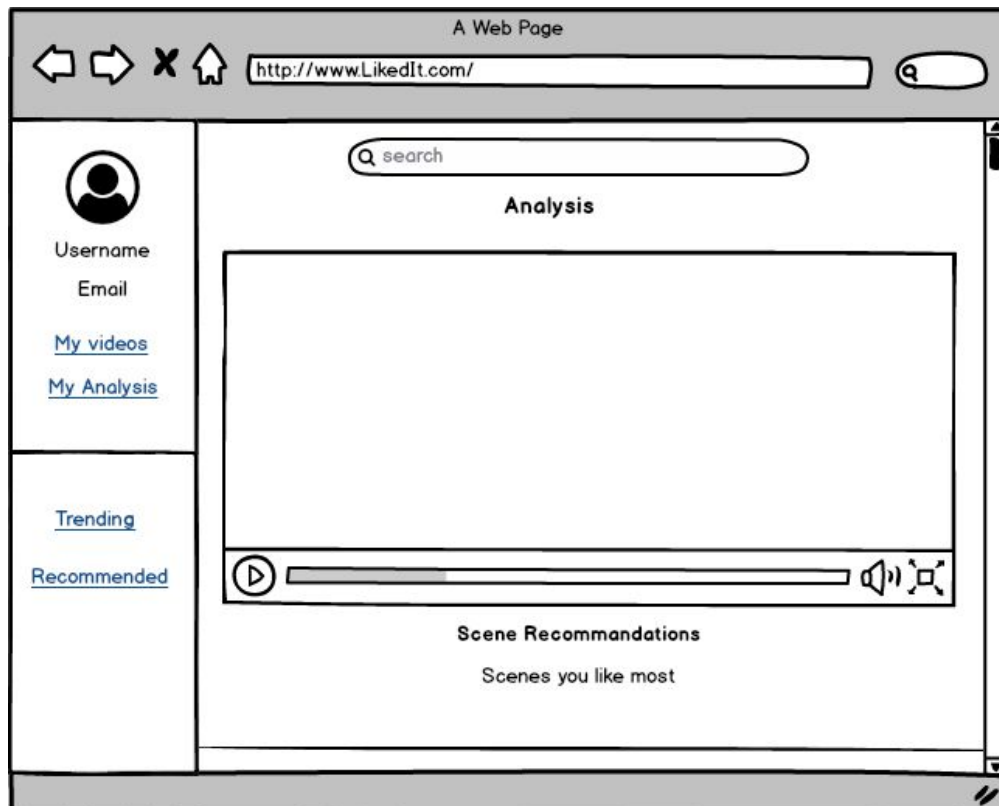


Figure 15: Recommended scenes

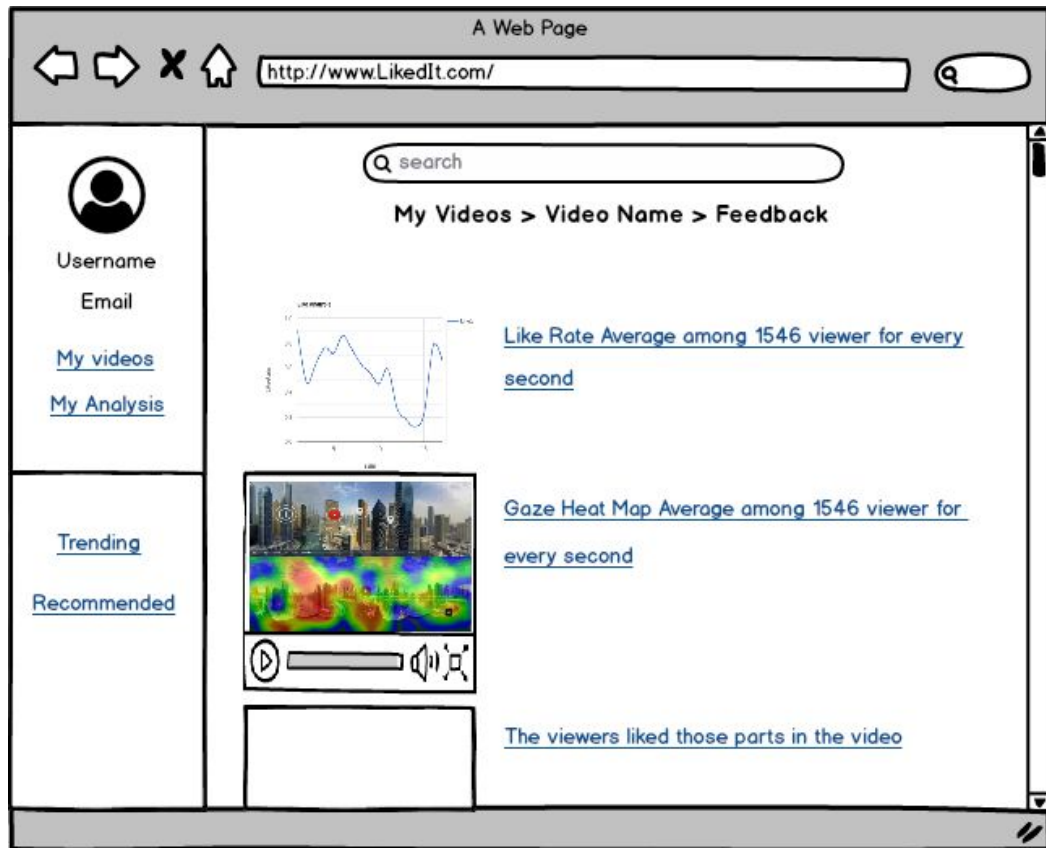


Figure 16: Feedback screen for the owner of a video

4. Other Analysis Elements

4.1 Consideration of Various Factors

In the process of analyzing, several factors that can affect the decisions within the development of LikedIt are thought and explicitly explained at the sections below:

4.1.1 Public Health

LikedIt is an application that requires a digital platform which results in the usage of digital screens like the computer screen. This may result in harmful effects: “at more than two hours of daily screen time can create a significant risk for problems with inattention and impulsivity” [7]. That’s why LikedIt should give a warning of over usage after 2 hours, saying that the user should close the application and rest for a while.

4.1.2 Public Safety

LikedIt should keep private information out of reach. Since the credentials of users are kept secure within the Google Account system, that is out of consideration. But the recommended videos list, analysis, and feedbacks should be private to users. In the feedbacks, LikedIt shouldn't give the names of users who watched the video, only the results of the analysis.

4.1.3 Public Welfare

LikedIt will not be affected by public welfare factors. Since it has no potential threats against public welfare.

4.1.4 Global Factors

LikedIt should support a trends list which can be adjusted to the trends of the world so that user would much likely to see globally popular videos. This way, users can follow the globally related and important trends of world.

4.1.5 Cultural Factors

LikedIt should support a trends list which can be adjusted to the trends of the user's country so that user would much likely to see their own culture related videos in their newsfeed.

4.1.6 Social Factors

In order to increase the social interactions of the users, LikedIt supports a sharing system which enables users to share their analyzes to other people.

4.1.7 Environmental Factors

LikedIt will not be affected by environmental factors. Since it's a software application that has no potential threats against environment.

4.1.8 Economic Factors

In order to make the application available to all users, LikedIt should be free of all charges.

Table 1: Factors that can affect analysis and design.

	Effect level	Effect
Public health	7	Warning for overexposure of screen time
Public safety	8	Securing the private data
Public welfare	0	None
Global factors	3	Global trends list
Cultural factors	4	Country trends list
Social factors	5	Sharing of analyzes system
Environmental factors	0	None
Economic factors	2	Zero cost for application

4.2 Risks and Alternatives

One of the features of LikedIt is that it measures the appreciation rate of users. In order to do that, we have to train a neural network model that requires a significantly high number of videos. We haven't been able to find a direct dataset for this specific purpose so that we will collect our data. This process contains a risk of not getting enough data to appropriately train our model which results in a biased model.

In such a case, since our model wouldn't be useful, we'd have to remove this feature. But in such a case, instead of this feature, we will include another feature, that is measuring which emotions the user has at which time and at which rate. Dataset for this feature already exists and wouldn't be as hard as to implement since collecting a new dataset wouldn't be the matter. This feature also can be added if we'd have enough time after implementing the appreciation rate feature.

There's another risk, which is about the network server. Since LikedIt analyzes the face of the user in real-time, sending the data to a cloud system to process the data and receive it may cause a delay. In such a case, instead of using an external cloud server to process the data, the application itself will process it but this may cause the LikedIt to not extendible to mobile phones for future features since their processors are not as powerful as computers.

Finally, there's another risk about cloud system which is that we couldn't find an Amazon Web Server service for free since some amount of free service is given to Bilkent computer engineers for their senior projects. In such a case, again we will have to use internal processors.

Table 2: Risks

	Likelihood	Effect on the project	B Plan Summary
Risk of not collecting dataset	3	Removal of appreciation rate analyzing feature	Adding another feature which already has dataset
Risk of Cloud Connection	6	Potential removal of mobile usage of system	Using an internal processor to process the data
Risk of AWS subscription	8	Potential removal of mobile usage of system	Using an internal processor to process the data

4.3 Project Plan

Table 3: List of work packages

WP#	Work package title	Leader	Members involved
WP1	Analysis	Zeynep Ayça Çam	Everyone
WP2	Design	Zeynep Hande Arpakuş	Everyone
WP3	Development 1	Elif Beril Şayli	Everyone
WP4	Development 2	Zeynep Nur Öztürk	Everyone
WP5	Testing	Muhammet Said Demir	Everyone

WP 1: Analysis			
Start date: 1 September 2019 End date: 11 November 2019			
Leader :	Zeynep Ayça Çam	Members involved:	Zeynep Hande Arpakuş Zeynep Nur Öztürk Elif Beril Şayli Muhammet Said Demir
Objectives: Objective of this work package is to analyze the project. Find out what the requirements are, determine the specifications,, overall management of the project etc. This is the first step of the project that gives us information about what to do and when to do.			
Tasks: Task 1.1 Requirement Meetings: To determine project's requirements. Task 1.2 Communication with Stakeholders: To determine what is needed in case of customer and what can be done by developers. Task 1.3 Specification Report: To determine the specification of the project Task 1.4 Analysis Report: To determine the mockups, use cases and plan the project Task 1.5 Manage the Project: To Determine when to do what to do and make proper work share			
Deliverables			

D1.1: Specification Report			
D1.2: Analysis Report			
WP 2: Design			
Start date: 21 September 2019 End date: 14 February 2020			
Leader :	Zeynep Hande Arpakuş	Members involved:	Zeynep Ayça Çam Zeynep Nur Öztürk Elif Beril Şayli Muhammet Said Demir
<p>Objectives: Objective of this work package is to make overall design of the project. Including the interface, high and low levels of project, database and the external data collector application. This part enables us to learn and determine the overall design of project elements.</p>			
<p>Tasks:</p> <p>Task 2.1 Interface Design : Design the interface for customer</p> <p>Task 2.2 Data Collector Design : To collect data we need a Data Collector app. To do this, we need a design to implement the properties.</p> <p>Task 2.3 High Level Design : During high level design, we define design goals of the project, and decompose the system into smaller subsystems that can be realized by individual teams (subgroups). Also select strategies for building the system, such as the hardware/software platform on which the system will run, the persistent data management strategy, the global control flow, the access control policy, and the handling of boundary conditions.</p> <p>Task 2.4 Low Level Design : Extent and validity of the design principles that were used to carry out this phase of the project must be explained in detail</p>			
<p>Deliverables</p> <p>D2.1: Mock-ups</p> <p>D2.2 : Data Collector Design</p> <p>D2.3 : High Level Design Report</p> <p>D2.4 : Low Level Design Report</p>			
WP 3: Development I			
Start date: 15 October 2019 End date: 1 February 2020			
Leader :	Elif Beril Şayli	Members involved:	Zeynep Ayça Çam Zeynep Nur Öztürk Zeynep Hande Arpakuş Muhammet Said Demir
<p>Objectives: Objective of this work package is to develop the first part of the project, the main elements. This part includes the external application to collect data, integration of the module that will scan the faces and implementation of neural network trained model. This part will be the main implementation of main elements of the projects. Later, we will use these parts to combine.</p>			
<p>Tasks:</p> <p>Task 3.1 Data Collector System : Developers will implement an application to get the needed data from users. This data which contains users' reactions to videos and their result.</p> <p>Task 3.2 OpenFace Integration : Implementing OpenFace library to process every data coming from data collector system and transforming them into something we can train model with.</p> <p>Task 3.3 Model Training : With the processed data coming from OpenFace, training a neural network model that will link the state of the face and appreciation rate.</p>			
<p>Deliverables</p> <p>D3.1: Data Collector App</p>			

WP 4: Development 2			
Start date: 1 February 2020 End date: 21 April 2020			
Leader :	Zeynep Nur Öztürk	Members involved:	Zeynep Ayça Çam Elif Beril Şaylı Zeynep Hande Arpakuş Muhammet Said Demir
<p>Objectives: Objective of this work package is to develop the second part of the project. This part includes implementation of secondary elements like cloud and processing there and to combine all the elements in a website. Also, to use these information to give a kind of feedback we intended to. Also the other elements like sharing, youtube integration. This part will be the end of development.</p>			
<p>Tasks:</p> <p>Task 4.1 Cloud Development : In our project major part will be done in the cloud to make real time analysis. Therefore, to analyze and give feedbacks the cloud needed to developed.</p> <p>Task 4.2 Cloud Integration : We need to merge the cloud and OpenFace to continue other parts.</p> <p>Task 4.3 Website Development : Because our project will be a website, we need to develop a back-end and front-end of a website.</p> <p>Task 4.4 YouTube Integration : Our program will get videos from Youtube and integrate them to our program with users' own account. We need to integrate this to our program.</p> <p>Task 4.5 Feedback Mechanism Development : In this part, we will prepare analysis system with the information from trained model.</p> <p>Task 4.6 Sharing System : On this part, sharing system for analysis will be implemented.</p>			
<p>Deliverables</p> <p>D4.1: Website of LikedIt</p>			
WP 5: Testing			
Start date: 7 November 2019 End date: 7 May 2020			
Leader :	Muhammet Said Demir	Members involved:	Zeynep Ayça Çam Elif Beril Şaylı Zeynep Hande Arpakuş Zeynep Nur Öztürk
<p>Objectives: Objective of this work package is to test the developed parts. This part will include the main element testings like models or data collector tests and overall system and cloud testing. This part will be the end of our project and if all the requirements met, project will end.</p>			
<p>Tasks:</p> <p>Task 5.1 Data Collector Testing : This part will test whether the data collector application works the way we intended or not. Whether it can record the video and upload it to dropbox correctly.</p> <p>Task 5.2 Model Testing : This part will test the model that we've trained. To see if it gives a biased result or a correct one.</p> <p>Task 5.3 Cloud Testing : This part will test the cloud system. To see if the output gives a correct information and don't lose any information in the network.</p> <p>Task 5.4 System Testing : This part is over all system testing. It will test whether the different elements of project integrated correctly or not.</p>			
<p>Deliverables</p> <p>D5.1: Overall LikedIt system will be delivered after this step.</p>			

4.4 Ensuring Proper Team-Work

In order for everyone to participate in leadership, we've divided the project into work packages and tried to make everyone the leader of a package. Also, in order for everyone to lead an equivalent load of work, we've divided some relatively big parts like development and gave them to different members. For each group member to take similar amount of work, we've assigned every member group to every work package so that everybody would know what is going on in these packages and wouldn't left behind.

Furthermore, if the work of an individual group member is done, she or Said will help the others members to do their parts as well in order to properly ensuring the equality of team work.

The commitments of the members will be uploaded on a privately shared GitHub repository that can be accessed by members and Jury Members at the link at the references [8].

4.5 Ethics and Professional Responsibilities

The application will hold data about users' appreciation of the videos. However, it will not hold any video or any photo of the user. So, our system will not violate the privacy of users. Only data to be shared will be the overall appreciation rate, age and country appreciation maps. Therefore, the owner of the video can't access the name of the user who likes the video or not. Personal data about appreciation to the videos with the name of the user used for recommending appropriate videos to the user and not shared with the third parties.

Furthermore, we will inform the users about why we need permission and we ask for their permission for such things as using the camera of the computer or the mobile device. We are also going to provide Terms and Conditions Agreement that the user needs to read before using the system which includes our privacy policy.

Lastly, during the implementation stage of our application, we want to avoid copyright infringement issues by giving priority to open source libraries.

On the other hand, if we have to use a library with copyrights, we are going to apply for the licensed version of the software library.

4.6 New Knowledge and Learning Strategies

In this project, since we'll be using a deep learning trained model, we have to learn neural networks and other things related to model training. For this purpose, we'll mostly use online learning strategy which we already discussed

and found online courses for these purposes. From Coursera we found a short but compact deep learning course [9]. From google courses, we found a related machine learning course that would also help us during the development process [10]. Since no of us taking machine learning or directly related courses at Bilkent, we have to learn through online platforms.

Other than this, since we'll be processing images of users in real-time that is taken from the video, we have to learn image analysis as well. For this purpose, 2 of us (Said and Zeynep Öztürk) takes image analysis course. Other than these, Said concurrently works on an augmented reality application at work which teaches him a thing or two about receiving, sending and streaming videos and connection.

YouTube will also be the most important tool to learn new technologies that we don't know and will be the place that will be the first visited place in each unknown method or technology [11]. Finally, for all kinds of bugs, StackOverFlow will be one of the most visited place for our project [12].

5. Glossary

OpenCv : OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products[13].

OpenFace : OpenFace is a Python and Torch implementation of face recognition with deep neural networks and is based on the CVPR 2015 paper FaceNet: A Unified Embedding for Face Recognition and Clustering by Florian Schroff, Dmitry Kalenichenko, and James Philbin at Google. Torch allows the network to be executed on a CPU or with CUDA [14].

6. References

- [1] “Predicting like-ratio on YouTube videos using sentiment analysis on comments”, 2019. [Online]. Available: <https://pdfs.semanticscholar.org/1ad2/ef79142ac7ecb83aa8a2d9c77876ddbc95a0.pdf>
- [2] “Sentiment analysis”, 2019. [Online]. Available: <https://monkeylearn.com/sentiment-analysis/>
- [3] “Open Face”, 2019. [Online]. Available: <https://github.com/TadasBaltrusaitis/OpenFace>
- [4] “OpenFace: an open source facial behavior analysis toolkit”, 2019. [Online]. Available: <https://www.cl.cam.ac.uk/research/rainbow/projects/openface/wacv2016.pdf>
- [5] “Getting Started With Microsoft Azure Cognitive Services - Emotion APIs”, 2019. [Online]. Available: <https://www.c-sharpcorner.com/article/getting-started-with-microsoft-azure-cognitive-services-emotion-apis>
- [6] “Using OAuth 2.0 to Access Google APIs ”, 2019. [Online]. Available: <https://developers.google.com/identity/protocols/OAuth2>
- [7] “Potential dangers of too much screen time for children ”, 2019. [Online]. Available: <https://www.news-medical.net/news/20190530/Potential-dangers-of-too-much-screen-time-for-children.aspx>
- [8] “LikedIt”, 2019. [Online]. Available: <https://github.com/handeArpakus/LikedIt>
- [9] “Neural Networks and Deep Learning”, 2019. [Online]. Available: <https://www.coursera.org/learn/neural-networks-deep-learning>
- [10] “Machine Learning”, 2019. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/ml-intro>
- [11] “Youtube”, 2019. [Online]. Available: <https://www.youtube.com/>
- [12] “StackOverFlow”, 2019. [Online]. Available: <https://stackoverflow.com/>
- [13] “OpenCv”, 2019. [Online]. Available: <https://opencv.org/about/>
- [14] “OpenFace”, 2019. [Online]. Available: <https://cmusatyalab.github.io/openface/>