



Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: LikedIt

High-Level Design Report

Zeynep Hande Arpaş

Zeynep Ayça Çam

Muhammet Said Demir

Zeynep Nur Öztürk

Elif Beril Şayli

Supervisor: Hamdi Dibekliolu

Jury Members: Özcan Öztürk, Selim Aksoy

1. Introduction	4
1.1 Purpose of the system	4
1.2 Design goals	5
1.2.1 Security	5
1.2.2 Usability	5
1.2.3 Cost	6
1.2.4 Performance	6
1.2.5 Extendibility	6
1.2.6 Marketability	6
1.3 Definitions, acronyms, and abbreviations	7
1.4 Overview	7
2. Current software architecture	8
2.1 Facial Analysis and Gaze Estimation	8
2.2 Youtube Data API	9
2.3 OAuth 2.0 to Access Google APIs	9
3. Proposed software architecture	9
3.1 Overview	9
3.2 Subsystem decomposition	11
3.3 Hardware/software mapping	14
3.4 Persistent data management	15
3.5 Access control and security	15
3.6 Global software control	15
3.7 Boundary conditions	16
3.7.1 Initialization	16
3.7.2 Termination	16
3.7.3 Failure	16
4. Subsystem services	17
4.1 Presentation Tier	17
4.1.1. Web View	17
4.1.1.1. UI Manager:	17
4.1.1.2. Home Page:	17
4.1.1.3. Video View:	17
4.1.1.4. Analysis Page:	17
4.1.1.5. Profile Page:	17

4.1.2. Controller	18
4.1.2.1. Controller:	18
4.1.2.2. Record Controller:	18
4.1.2.3. Account Controller:	18
4.1.2.4. Server Controller:	18
4.1.2.5. Video Controller:	18
4.1.2.6. Database Controller:	18
4.1.2.7. Analysis Controller:	19
4.2. Application Tier	19
4.2.1. Application Manager	19
4.2.1.1. App Manager:	19
4.2.1.2. Database Manager:	19
4.2.1.3. Connection Manager:	19
4.2.1.4. Analysis Manager:	19
4.2.1.5. Cloud Manager:	19
4.3. Database Tier	20
5. New Knowledge Acquired and Learning Strategies Used	20
6. Glossary	22
7. References	23

1. Introduction

Nowadays, human-computer interactions have been increasing rapidly. This interaction creates new social environments for people to share their lives. "With the advent of mobile applications and social websites such as YouTube, Vine, and Vimeo, we have observed an increase in the number of online videos shared by people expressing. To give you a better idea of how popular these websites are, more than 300 hours of video is uploaded to YouTube every minute." [1]. The amount of users in these platforms increases attention from companies, clients, and researchers. Creating and influencing people's experiences has become a valuable differentiation strategy for the owner of videos. Therefore, research data from these platforms are necessary and attractive in today's world.

The purpose of our Senior Design Project in Bilkent University Department of Computer Engineering is to contribute research about human-computer interaction when people watch online videos. We will capture physiological reactions in real-time to accurately research how humans appreciate. One of the strongest indicators for an understanding of appreciation is the human's face. The facial expression represents one of the most important non-verbal means of communication.

This application provides a feedback report to the video owner about how much the other users liked or disliked the video scene by scene, where do the users look at in the video mostly, which groups liked their video mostly. The application can be used by every age, gender, and ethnic group who watches or records videos. Furthermore, in our application, we recommend videos that users might like in the future based on their reactions to the videos they watched.

1.1 Purpose of the 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 [2]. 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 [3]. 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. 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.

1.2 Design goals

1.2.1 Security

- The application should not keep the data on 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.

1.2.2 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.

1.2.3 Cost

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

1.2.4 Performance

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

1.2.5 Extensibility

- 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.

1.2.6 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.

1.3 Definitions, acronyms, and abbreviations

Term	Definitions, acronyms, and abbreviations
Open Face	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[4].
MySQL	MySQL is an open-source relational database management system. It is based on the structured query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL[5].
Node.js	Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser[6].
OAuth 2.0	Google API for authentication and authorization [7].
Facial landmarks	Facial landmarks are used to represent the facial areas.
Gaze vectors	The vector between the pupil center and the corneal reflections can be used to compute the point of regard on the surface or the gaze direction [8].

1.4 Overview

LikedIt is a website to watch videos. After users log in, 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 [7]. 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.

2. Current software architecture

Although there aren't any application which gives feedback about liking rates when users watch videos, there are several applications and libraries that provide facial analysis and gaze tracking. This section is to reveal these modules and their functionalities.

2.1 Facial Analysis and Gaze Estimation

OpenFace which is open-source platform can operate on real-time data video feeds from a webcam, recorded video files, image sequences, and individual images. It is possible to save the outputs of the processed data as CSV files in case of facial landmarks, shape parameters, action units and gaze vectors [9]. Facial landmark detection has great importance to make the facial analysis.

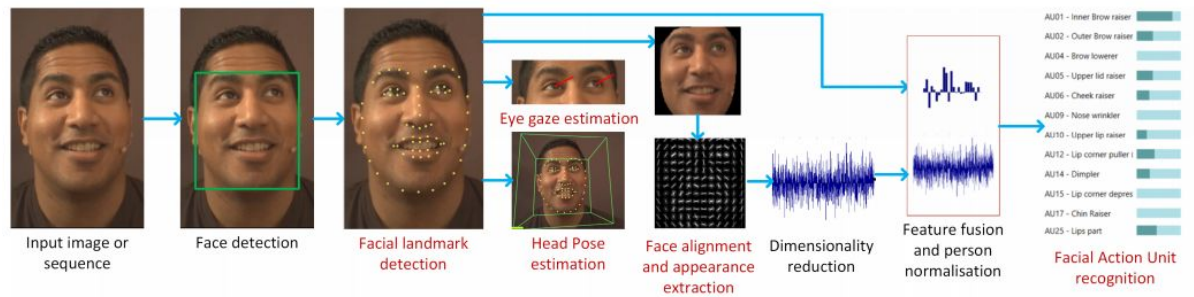


Figure 1: Open Face results [9]

To provide gaze tracking information about videos, we use eye gaze estimation module of OpenFace.

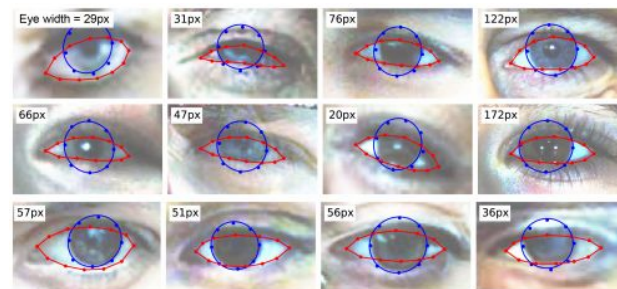


Figure 2: Gaze Tracking [9]

2.2 Youtube Data API

Youtube is a video-sharing platform [10]. The main source of videos in our program is Youtube. Also, we use subscriptions of users of Youtube via their Google account. Youtube provides YouTube features to integrate our program with the YouTube Data API. These features are like including the ability to upload videos, searching video and more. The main purpose of using this API is obtaining information of users, taking videos and their corresponding ids.

2.3 OAuth 2.0 to Access Google APIs

Users can only log in to the system with their Google account. Google provides OAuth 2.0 to Access Google APIs for the authorization process [7]. The program requests an access token from the Google Authorization Server and takes token via Google API.

3. Proposed software architecture

3.1 Overview

In this part, we will firstly explain subsystem decomposition, where the subsystem structure of our system is described in detail with diagrams and the classes. Then, we will provide hardware/software mapping of the system which shows the allocation of resources in our project. Then we will explain the persistent data management. We will also explain access control and security which defines the access boundaries of the users and security management in our system. Then we will explain global software control with the illustration of general flow in our system. Finally, in our boundary conditions, we will mention the initialization, termination, and failure conditions.

3.2 Subsystem decomposition

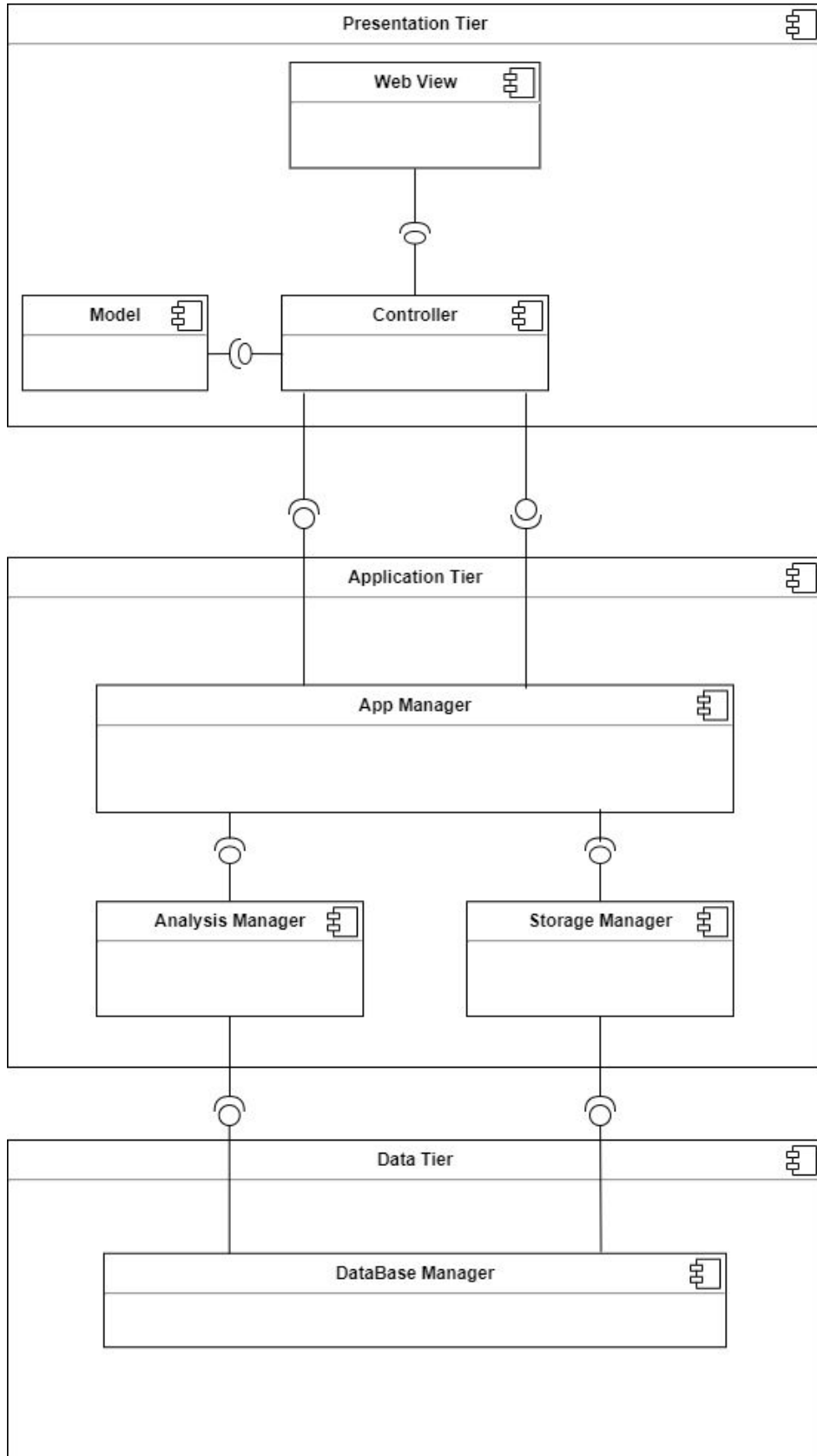


Figure 3: Decomposition Diagram

Our system is formed with a 3 Tier Decomposition. These tiers are presentation tier, application tier, and data tier. This tier decomposition gives us the flexibility to update project since tiers are independent of each other

The presentation tier consists of user interfaces. Web view, controller and model packages are in this tier where they have different classes under each of them. You can see details of web view, controller and application manager from below diagrams:

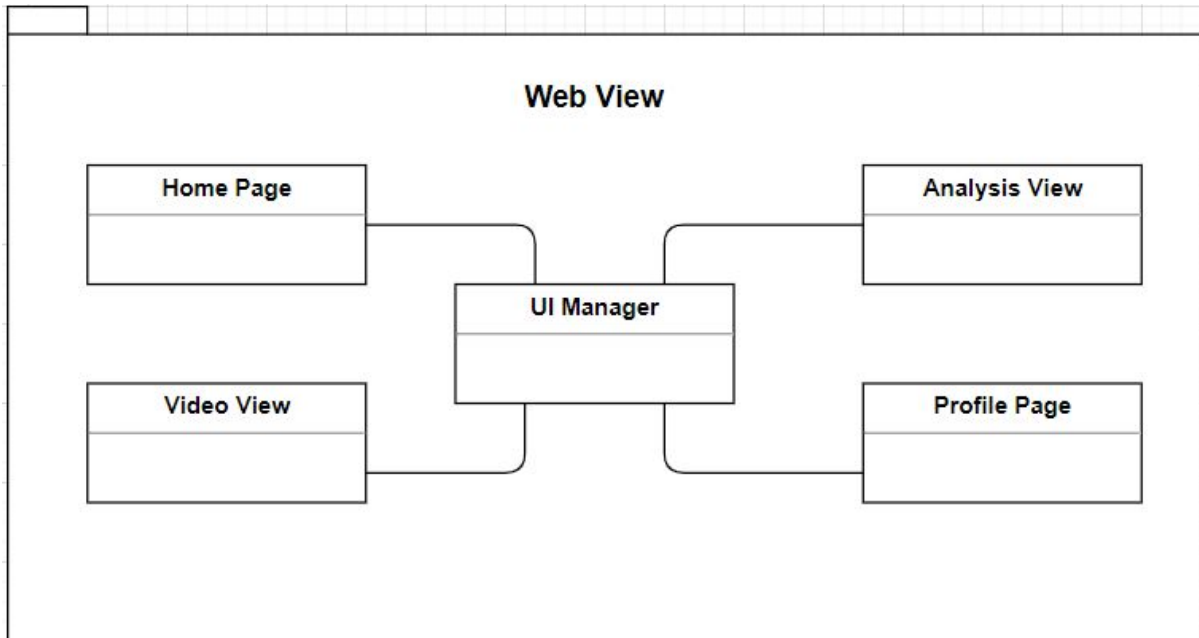


Figure 4: Web View

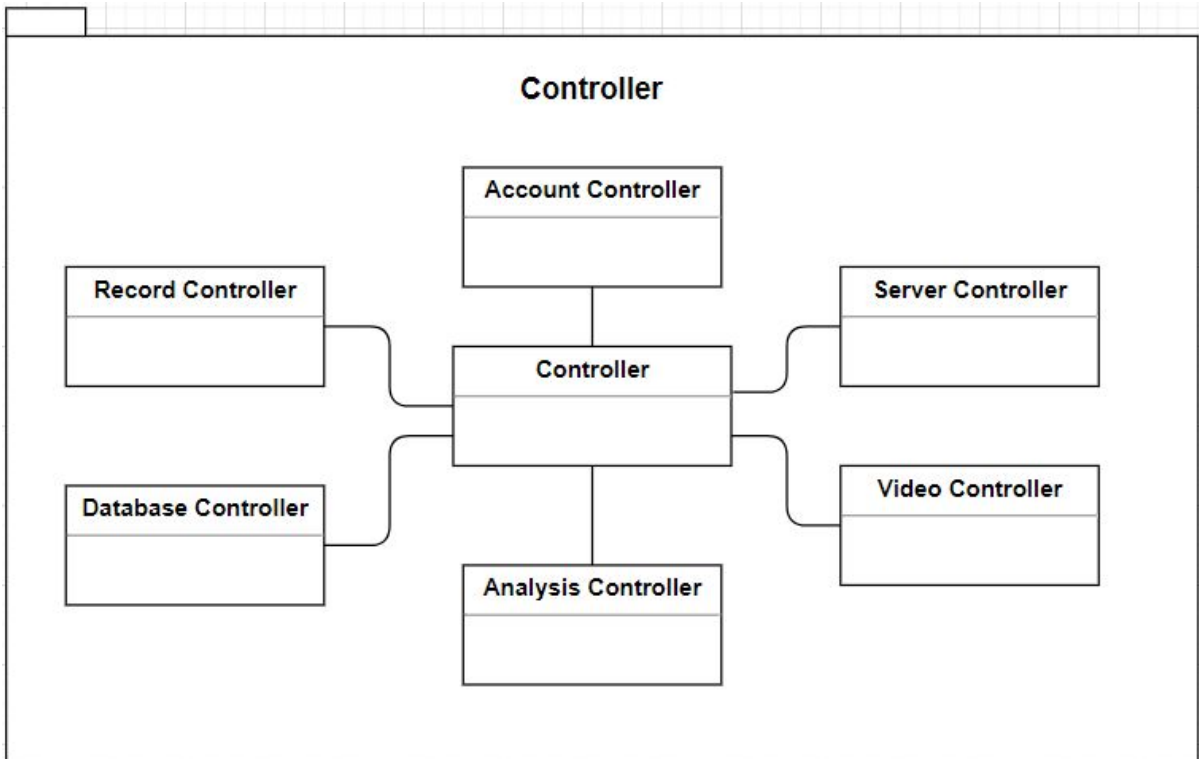


Figure 5: Controller

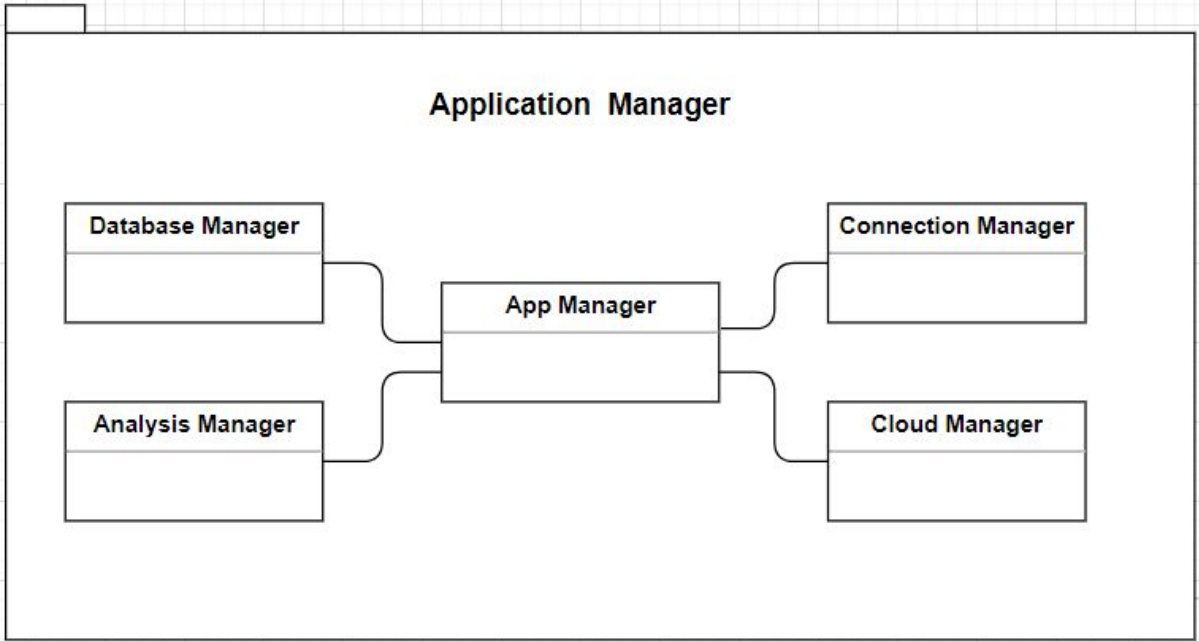


Figure 6: Application Manager

3.3 Hardware/software mapping

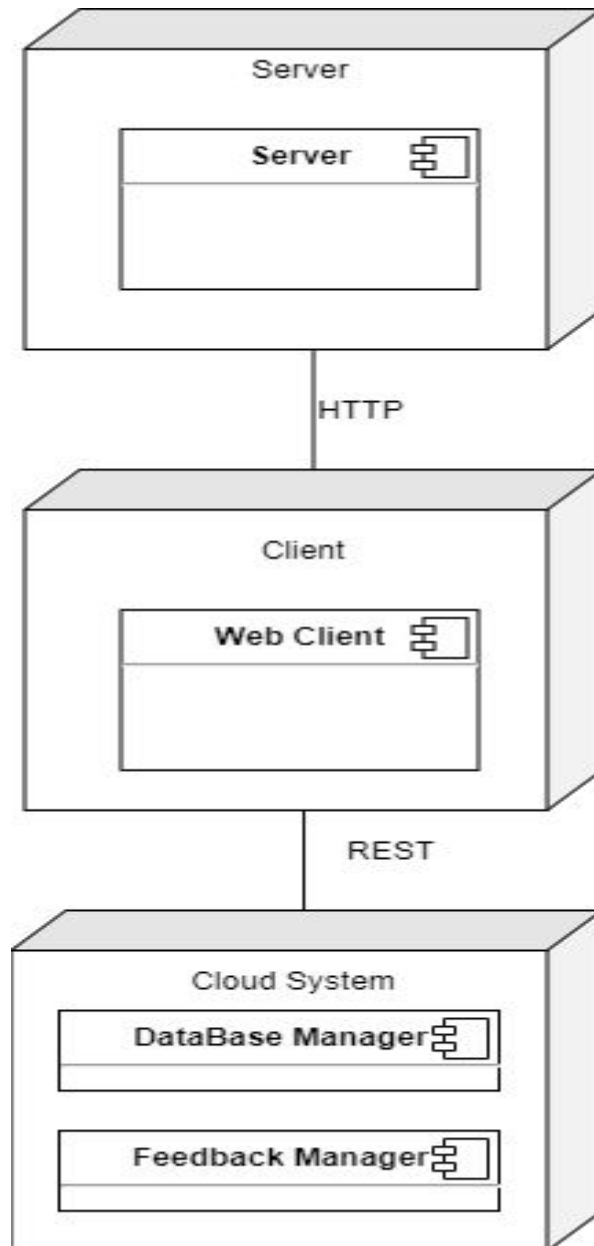


Figure 7: Hardware/Software mapping

The tiers of the system will be divided into server, client and cloud system. The system has one client, which is a web client. These clients will connect to the server with an HTTP protocol with TCP. The web client will be responsible for the presentation layer of the system. The server software will be in the same machine with the client and will be responsible for running necessary models on the background if necessary. The cloud tier will have 2 different components. One of them will work as a database component and stores user data and application data.

The second part will for other programs and models to run in order to provide feedback for users.

3.4 Persistent data management

In the LikedIt system, some data need to be persistent, having a longer lifetime than a single execution. One way to have a persistent object is to store it inside a Database. We are going to use Google's Database Service as a relational database to store persistent object information. By using Google Database, we ensure ease of database migration and integrity. One of the Database engines is MySQL, Oracle, and Microsoft SQL Server. We'll make our system fast by providing "index" like structures on our database. We'll index the mostly retrieved data attributes in our Database. Such data might be considered as the login and profile credentials of the users.

Video Analysis results can be considered as another type of persistent data object. However, in our system, we will not hold processed videos for every processed record. For every video, we will only store some of the user's processed video results in text form. LikedIt will not store the videos of the user without permission.

3.5 Access control and security

For LikedIt, everyone who has a Google account can access the web site. Users log in the website with their Google username and password. The system checks this information with Google API, therefore, the system doesn't store them. After signing in, each user can update their profile settings and see analyze results. For accessing, LikedIt's only requirement is Google account.

LikedIt provides data privacy for its users. For security, while an analyzed result of a video is storing for the owner of it, the names of the viewers are hiding. Also, the information about which user watch which video is just stored in the database. Because the system doesn't record the user's video, the only sensitive data is this video information in the system's database.

3.6 Global software control

Users will have to have a Google account to use LikedIt. Users can access our website as an uploader and watcher. Like Youtube, we don't distinguish according to user types. They will be able to register with one

Google accounts addresses by providing necessary information and start using LikedIt. After the sign-in process, the user can record his/her video or view sample videos offered by our system. These processes are controlled via the video controller.

The client-server communication, video controller starts the communication and delegates partial control to the server-side with the cloud controller. We offer two options to users. Users can choose to send videos to the cloud and real-time data processing starts in there. If users don't want to send his/her video to the server, we can process videos in the client host and send its result data to our cloud. Cloud controller controls all server-side jobs including data retrieval from database or processing video by using main modules.

Community settings and insights will be accessible to authorized users (admins) only which are creators of the website. Individual settings can change with specific users. The rights of the accessibility of stored videos belong to only.

3.7 Boundary conditions

3.7.1 Initialization

Users must be using the web application of LikedIt and must have a Google account to sign in.

3.7.2 Termination

Users can terminate the app by logging out or closing the browser window/tab.

3.7.3 Failure

While using the system, the user may face some errors such as log-in error, internet connection error, face recognition error, etc. Our goal is to minimize these errors and give the users as much information as possible about the cause of failures.

4. Subsystem services

4.1 Presentation Tier

4.1.1. Web View

Web View subsystem shows the visual interaction between the system and the user. This system will be implemented using HTML, Javascript/PHP, MySQL, Cloud. Individual explanations of the components of the view module are as follows:

4.1.1.1. UI Manager:

This component controls the web view subsystem which controls the distributions of tasks like opening the home page then passing to the profile page, analysis page, etc.

4.1.1.2. Home Page:

When the user enters the site directly opens the home page. From this point, every interaction between the user and the site is controlled by this page until the user leaves this page. User can see the recommended videos based on her/his real likes, trending videos, subscribed channels videos. Also, the user can be directed to the profile page, video view page, analysis page.

4.1.1.3. Video View:

When the user clicks on a video, this page is opening. On this page, video is in Theater Mode like in Youtube's theater mode. When the user is on this page, Video view page will control the user-system interaction. Users can be directed to the profile page, the home page, analysis page.

4.1.1.4. Analysis Page:

When this page is opened by the user this page manages the interactions between the user and the system, until it leaves this page. On this page, user will see the output of his/her results. From this page, the user can be directed to the profile page, view video page, the home page.

4.1.1.5. Profile Page:

When this page is opened by the user this page manages the interactions between the user and the system, until it leaves this page. Here, the user can see his/her analysis results, his/her uploaded results. From this page,

the user can be directed to the analysis page, view video page, the home page.

4.1.2. Controller

4.1.2.1. Controller:

This is the class controls and regulates the communication between other classes like starting the records, manage the accounts, connecting the site to the server, etc.

4.1.2.2. Record Controller:

This class handles the recording of the user as synchronized. When the user starts the video, this class needs to handle the recordings with starting of the video and should handle the pauses.

4.1.2.3. Account Controller:

This class handles the actions towards the user's account. This includes signing up the user, checks the signing in, and changing the password issues.

4.1.2.4. Server Controller:

The class handling the action to connect to the server with system and maintaining the connecting in the time period that the video will be uploaded to server and processed video and analysis data will be uploaded to the system.

4.1.2.5. Video Controller:

The class handling the actions of the client towards video showing to the user and before showing the video shows the user's camera view to the user to prepare himself for video recording.

4.1.2.6. Database Controller:

The class handling the connection between the Google Cloud database server with the site. For instance, all the information needed for analyzing the user's webcam video, the result of OpenFace needed to be sent to the database, and these results needed to be calculated in the cloud and results will be entered into a database and this will return to the analysis controller.

4.1.2.7. Analysis Controller:

The class handling the analysis of user videos. Inside the server calculates the outputs and sends these results to the cloud, by database controller

these results entered to the database. After the results are calculated, Analysis Controller receives the cloud results and shows it to the user.

4.2. Application Tier

This provides the main functionality of the system. It connects the Presentation Tier and Data Tier.

4.2.1. Application Manager

4.2.1.1. App Manager:

The app manager handles the requests that are coming from the client-side (presentation tier) like the user enters a page and the app manager does the works beside the UI.

4.2.1.2. Database Manager:

It provides the connection between the database and the application. It gets information from the database and sends them to the site and from the site gets the results and sends it to the database tier.

4.2.1.3. Connection Manager:

It provides the connection within the site and cloud groups in the system. It is responsible for managing of sending/receiving inputs/outputs. It is also responsible for creating a connection with OpenFace if it works in the backend.

4.2.1.4. Analysis Manager:

It provides the analysis of the project, gets the results from the server, calculates the results and turns back to the presentation tier. While calculating it uses the Cloud's trained data.

4.2.1.5. Cloud Manager:

It provides the cloud storage for sending/receiving files within the site and cloud group environments. This cloud includes a deep learning model. There is also a feedback mechanism to calculate the result for a given image result.

4.3. Database Tier

Data Tier Subsystem provides the data management of the system. It cannot directly connect with the presentation tier for security. It interacts with the Application Tier Subsystem only.

5. New Knowledge Acquired and Learning Strategies Used

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 an 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 [12]. From google courses, we found a related machine learning course that would also help us during the development process [13]. Since none 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 Nur Öztürk) takes an image analysis course. Also, for this real-time processing, we started to learn how to use cloud technology. Because of features and advantages, we focused on the Google cloud. Besides video processing, this technology will be used for accessing the database. So, we also practiced this field too. For this practicing, we followed Google's tutorial.

LikedIt is basically a video platform and these videos are pulled from YouTube and to log to the website, the user should use his Google account. For these features, the system is using APIs of Google and YouTube. So far, we learned how to use these APIs and integrate them into our program. Again for this, we benefited from Google's tutorials. Also, we used other tutorials too [11].

In our project, the system runs with OpenFace. So, the user should have OpenFace on the computer. To solve this, we learned how we can create an installation package for OpenFace and make communicate our website and the computer.

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 [10]. Finally, for all kinds of bugs, StackOverFlow will be one of the most visited places for our project [14]. Of course, in all these processes we always consult with our project advisor. He directs us about how we can improve our strategies and ideas, with his experiences and knowledge.

6. Glossary

-A-

Analysis, 3: 4: 6: 7: 14: 17: 18: 19

API, 6: 8: 14: 19

Application, 3: 4: 5: 7: 11: 12: 14: 15: 18

-C-

Client, 3: 13: 15: 17: 18

Cloud, 7: 13: 15: 16: 17: 18: 19

-D-

Database, 6: 14: 15: 17: 18: 19

-E-

Estimation, 7: 8

-F-

Facial, 3: 6: 7

-G-

Gaze, 8: 7: 6

Google, 6: 8: 14: 15: 17: 19

-L-

LikedIt, 4: 5: 6: 14: 15: 19

-M-

MySQL, 6: 14: 16

-N-

Node.js, 6

-O-

OpenFace, 6: 7: 8: 17: 18: 19

-S-

Server, 13: 15: 17: 18

-Y-

Youtube, 3: 4: 5: 6: 7: 8: 14: 16: 19

7. References

- [1] “Multimodal Sentiment Intensity Analysis in Videos: Facial Gestures and Verbal Messages” *Sentic.net*, 2019. [Online]. Available: <https://sentic.net/multimodal-sentiment-intensity-analysis-in-videos.pdf> [Accessed: 5- Oct- 2019].
- [2] “Predicting like-ratio on YouTube videos using sentiment analysis on comments”, 2019. [Online]. Available: <https://pdfs.semanticscholar.org/1ad2/ef79142ac7ecb83aa8a2d9c77876ddb c95a0.pdf> [Accessed: 5- Oct- 2019].
- [3] “Sentiment analysis”, 2019. [Online]. Available: <https://monkeylearn.com/sentiment-analysis/> [Accessed: 5- Oct- 2019].
- [4] “Open Face”, 2019. [Online]. Available: <https://github.com/TadasBaltrusaitis/OpenFace> [Accessed: 5- Oct- 2019].
- [5] “MySQL Definition”, *Techterms.com*, 2019. [Online]. Available: <https://techterms.com/definition/mysql#:~:targetText=MySQL%2C%20pronounced%20either%20%22My%20S%2DQ%2DL.modifying%20information%20i n%20the%20database> . [Accessed: 20- Dec- 2019].
- [6] “Node.js - Wikipedia”, *Wikizeroo.org*, 2019. [Online]. Available: <https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEu b3JnL3dpa2kvTm9kZS5qcw>. [Accessed: 25- Dec- 2019].
- [7] “Using OAuth 2.0 to Access Google APIs ”, 2019. [Online]. Available: <https://developers.google.com/identity/protocols/OAuth2>
- [8] “Eye tracking - Wikipedia”, *Wikizeroo.org*, 2019. [Online]. Available: <https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEu b3JnL3dpa2kvRXIX3RyYWNraW5n>. [Accessed: 31- Dec- 2019].
- [9] Baltrusaitis, Tadas, “OpenFace: an open-source facial behavior analysis toolkit”, 2016. [Online] Available: <https://www.cl.cam.ac.uk/research/rainbow/projects/openface/wacv2016.pdf>
- [10] “Youtube”, 2019. [Online]. Available: <https://www.youtube.com/>
- [11] “Accessing Data in an OAuth Server”, 2019. [Online]. Available: <https://www.oauth.com/oauth2-servers/accessing-data/>

[12] “Neural Networks and Deep Learning”, 2019. [Online]. Available: <https://www.coursera.org/learn/neural-networks-deep-learning>

[13] “Machine Learning”, 2019. [Online]. Available: <https://developers.google.com/machine-learning/crash-course/ml-intro>

[14] “StackOverFlow”, 2019. [Online]. Available: <https://stackoverflow.com/>