Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Here!

High Level Design Report

Mert Aslan, Rahmiye Büşra Büyükgebiz, Hakkı Burak Okumuş, Ozan Aydın, Yüce Hasan Kılıç

Supervisor: Abdullah Ercüment Çiçek
Jury Members: Çiğdem Gündüz Demir and Can Alkan

High Level Design Report
Dec 27, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491.

Contents

High Level Design Report

Project Short-Name: Here!

1 Introduction

The COVID-19 pandemic that struck the world in early 2020 led humanity to search for alternate solutions to everyday tasks. It affected people to the point that these tasks which would have been quite trivial in the past, now require immense care to complete. Along with these safety measures came the exercise of social distancing, which led to a complete overhaul of human to human interaction. We now communicate mostly online, with little to none face to face interaction. Video conferencing applications, such as Zoom, Google Meet or Microsoft Teams are much more prevalent than ever, due to this unprecedented circumstance. Even though this new type of communication is necessary to prevent the spread of the disease, it certainly has its drawbacks.

As students, during our time being educated remotely, we have experienced some problems that affected both the instructors and the students. It is evident that an online education setting has much less room for student-instructor interaction compared to a classroom setting. This mainly has to do with the fact that video conferencing applications such as Zoom or Google Meet are not specialized to be used as an education medium. Many valuable information, such as the attention of the students to the lectures, that were constantly gathered by instructors in classrooms are lost in online education due to the fact that these conferencing mediums are simply not specialized enough to gather such information. Our experiences show that the loss of such information leads to a complete detachment between the instructors and students, which overall diminishes the effectiveness of online education. Our application aims to overcome this problem, by providing a solution using machine learning and computer vision algorithms that would increase the amount of knowledge the instructors receive about their students both in real time and right after lectures.

Students also lose their ability to effectively follow the lessons, as these video conferencing mediums do not provide them any means to enhance their learning process. They are now essentially “participants” rather than “students” that do not have the essential tools they once had in their classrooms. This unfortunate situation distances them from the learning process itself. We also aim to solve this problem, by providing a platform specialized for the needs of the students in an online education environment.

In this report, we intend to provide a high level design of the system. First, the purpose of our system and its design goals will be discussed. Then, existing systems will be examined. Afterwards, we will discuss the system architecture of our project, we will provide subsystem decomposition and deployment diagrams and proceed to explain them further. We will then revisit considerations of various factors and talk about the effect on our design. The last section is devoted to teamwork details where we explain the collaboration and communication processes of our teamwork.

1.1 Purpose of the system

Here! is a web based virtual classroom environment that is designed to simulate the traits of a real classroom for both the instructor and the students. Here! is designed with the students' perspective in mind, and its main purpose is to provide a platform that specializes in education, with features that will assist everyone in the classroom. This aim includes combining computer vision and machine learning algorithms and basic educational tools that we feel are missing from the generic video conference programs to provide a stable, smooth and productive learning experience.

1.2 Design goals

1.2.1 Security

- The program must ensure that video recordings of users will not be shared with any other 3rd party application.
- Users must accept the terms of service and privacy policy to use the program.

1.2.2 Usability

- The program must be suitable for webcams with different resolutions.
- The program must be supported by different browsers.
- The program must be self-explanatory and user-friendly.

1.2.3 Performance

- The program must be working fast such that it can process all frames of a video captured in real time from 1 second to no delay.

1.2.4 Extensibility

- The program will be developed as a web application at first but it must be extensible for desktop, Android and iOS environments.

1.2.5 Scalability

- Since the focus area of our program is education it must be scalable to handle a large number of users from various institutions.
- The database and the server must be able to handle large numbers of concurrent video calls.

1.2.6 Reliability

- The program must detect motions accurately since mistakenly detecting a hand raise can interrupt the flow of the class.
- Similarly, missing a single one will affect satisfaction of the student.

1.3 Definitions, acronyms, and abbreviations

- **TA**: Teaching Assistant
- **UI**: User Interface
- **API**: Application Programming Interface
- **Server**: Houses the database and is the backend of the system. All logical operations on the data are done here.
- **DB**: Database
- **RTC**: Real Time Communication
- **HTTP:** Hypertext Transfer Protocol
- **AWS:** Amazon Web Services
- **EC2:** Elastic Computing
- **RDS:** Relational Database Services
- **S3:** Simple Storage Service

1.4 Overview

Here! is a video conferencing service that is specifically designed for educational needs of both students and instructors. It will include the general characteristics of a video conferencing application with the inclusion of practical educational tools. These tools can be divided into two parts, an online TA that will use motion and eye tracking technologies in order to analyze videos of students to come up with informative data about the class session, and other complementary features that will make online education more reliable and easy for both students and instructors.

If an instructor chooses to do so, it can use the online TA feature to analyze and gather data from videos. Our online TA will print out attendance rates by analyzing how frequently a student participates or raises their hand. It will also notify the teacher simultaneously when a student raises their hand or gives a specific reaction, like a thumbs up. Another feature is to track eye movements in order to understand if a student is distracted and notify the teacher during the lecture.
Other than our online TA, we have multiple features that support online education. One feature will be a simultaneous note taking, where you can instantly screenshot the current slide or shared screen and add that into your notes, where you can view them later in your profile with added personal notes from you. Another main feature will be the slide sharing, where students can navigate freely through a slide that the teacher is actively sharing in order to better read and understand the content without hindering the rest of the class, and sync up with the instructor whenever they want. Additionally, instructors will be able to share their screens separately from the slide share.

2 Current software architecture

Even though there are many programs that are being used for online education, most of them are conference apps, and their functionalities are limited to screen sharing. *Here!* will provide unique functionalities such as machine learning based teaching assistant that will track the attendance of students, warn them and provide statistics to instructors about students' attendance after the lectures. Moreover, there will be an interactive note taking system which allows students to take notes directly onto the slides that are shared by the instructors.

2.1 Similar apps and platforms

2.1.1 General Video Conference Programs (Zoom, Microsoft Teams, etc.)

Today, most of the schools use general purpose video conference programs in online lectures such as Zoom, Microsoft Teams, etc. However, these programs consist of basic video conference features like video and screen sharing, voice and text communication while they lack education specific features that we are planning to aim such as slide sharing, machine learning based teaching assistant, or interactive note taking system.

2.1.2 Google Classroom

- Instructors can add students to the lectures directly, or they can send join links to students.
- They can set up a class and create class work that appears on students' calendars.
- Instructors are able to grade students consistently with rubrics integrated into student work.
- They can see originality reports of the student works for potential plagiarism.

2.1.3 Vedamo

- Instructors can create their own academy to manage their courses, content and learners.
- Users can keep records of online lectures.

2.1.4 Blackboard

- Instructors can keep track of the students’ individual activity during the lecture time relative to others.
- They can compare the change in overall attention between different class hours.
- Users can easily keep track of their data from the mobile application.

2.1.5 Tophat

- Instructors can create polls, quizzes and synchronous discussions to encourage student participation.
- They can record class meetings and share videos directly from the application.
- They can edit the slides that they already uploaded through the application.

2.1.6 BigBlueButton

- BigBlueButton provides a live multi-user whiteboard for both instructors and students.
- People can zoom, highlight, draw and write on the presentations through the application.

3 Proposed software architecture

3.1 Overview

Here! will consist of two main layers, the first one is the client side where the users will be able to launch the web application in order to join the session. The other one is the server side which handles all the comparisons, computer vision algorithms and data retrieval from the database. The server will be configured using Amazon Web Services (AWS). The code will run on an AWS EC2 machine, while the database will be stored using AWS RDS. Last but not least, AWS S3 will be used in order to store large files, such as lecture notes as PDF files. The tables stored in AWS RDS will point to these files in AWS S3 in order to access them upon a student request. The client and server layers will communicate with each other through sockets and HTTP requests, handled by various controllers.

3.2 Subsystem decomposition

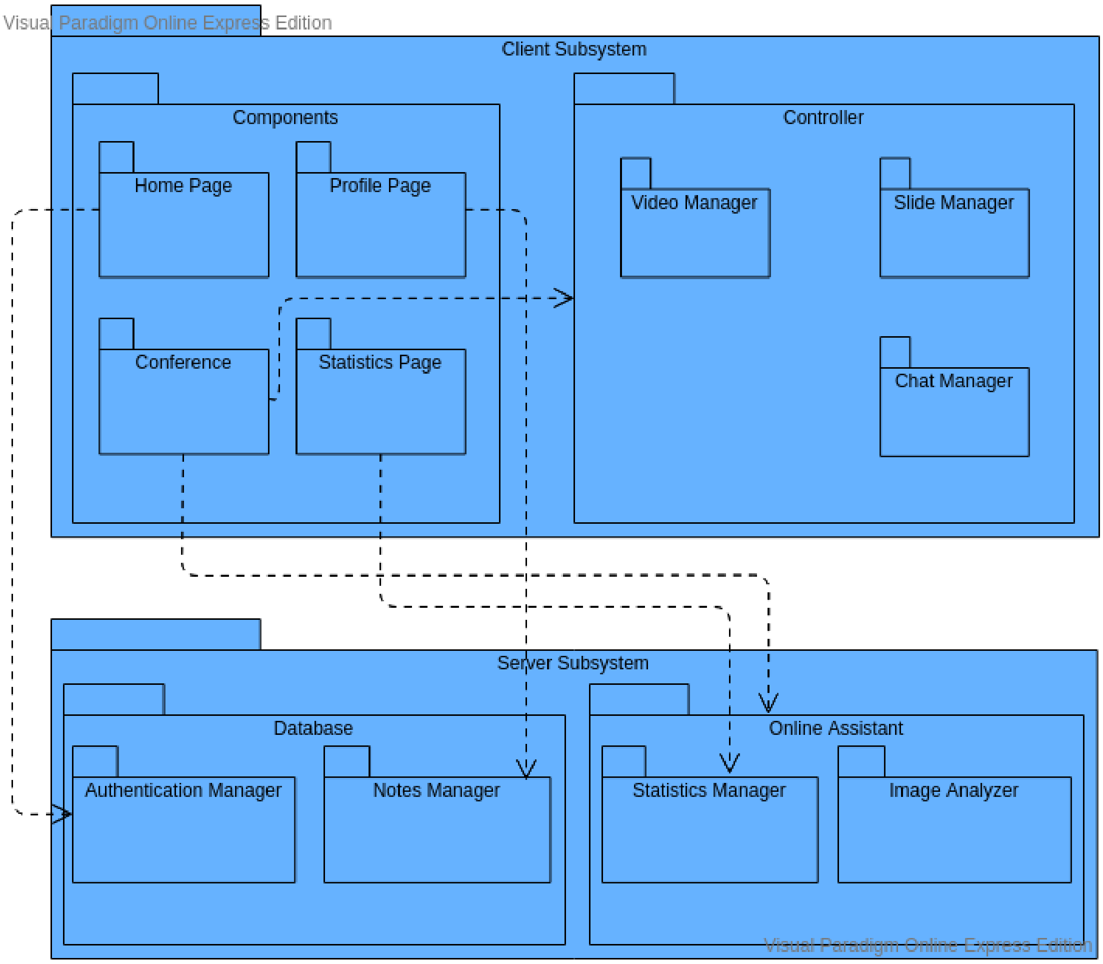

***Figure 1:** Subsystem Decomposition Diagram*

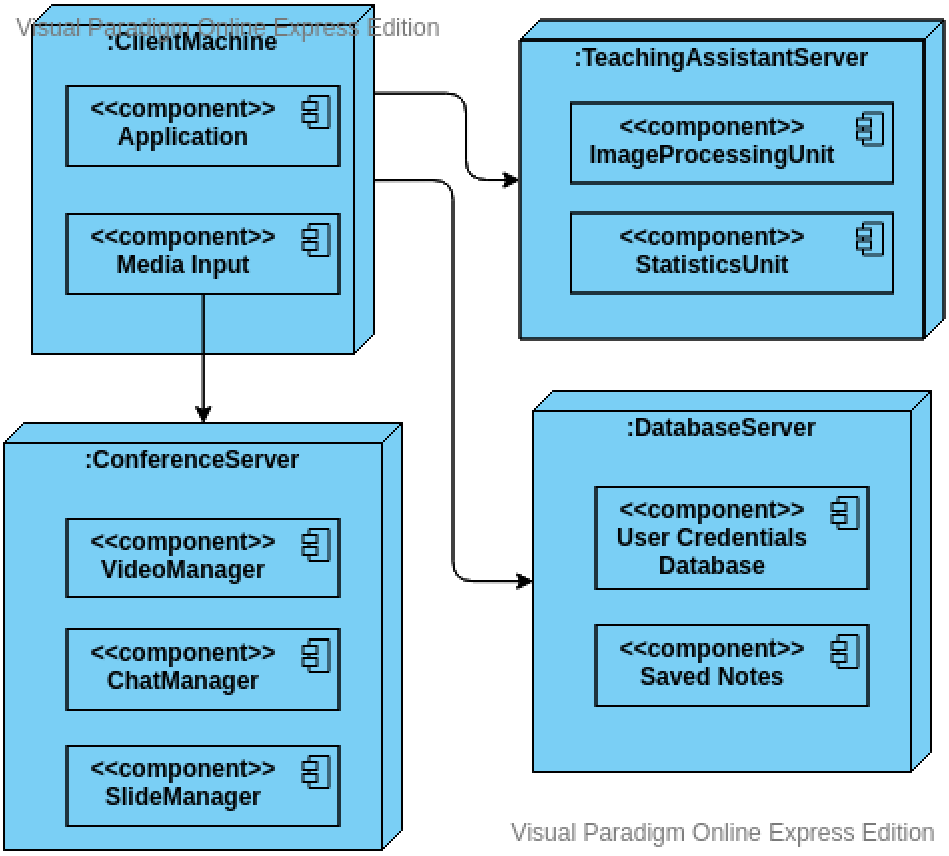
:ClientMachine
<<component>> Application
<<component>> Media Input
:TeachingAssistantServer
<<component>> ImageProcessingUnit
<<component>> StatisticsUnit
:ConferenceServer
<<component>> VideoManager
<<component>> ChatManager
<<component>> SlideManager
:DatabaseServer
<<component>> User Credentials Database
<<component>> Saved Notes

Figure 2: *Deployment Diagram*

3.3 Hardware/software mapping

Here! will be a web application developed with Angular. It will use media devices from the client's computer such as the camera and microphone. The program will capture the video as frames and convert this analog input into digital representation to send the information to the teaching assistant server which will be hosted through Amazon EC2 to analyze the gestures and movements of students to be able to give them feedback. The program will depend on memory, network adapters and processor units of the client’s device for running the application on the browser and sending/receiving data through the lectures. Teaching assistant of the program will run on a server that we assign and it will be dedicated for machine learning purposes, namely analyzing the video data that the program receives from the client. User credentials and notes that students saved during lecture will be stored in a database server and will be accessed from application when requested. Client’s machine will be communicating throughout the conference server to send the media input to the server while buffering media from other students and the instructor. Conference server will synchronize the media inputs and provide real time audio, video and text communication.

3.4 Persistent data management

Video and audio inputs are the major data components for the application. Transfer and synchronization will be handled by WebRTC framework. Here! will also use the video footage of students to track their attention during lectures. Since this will be a real-time analysis, persistent data flow between server and client will be crucial to keep the program working effectively. We will try to establish this stable connection between the teaching assistant server and client’s machine using the Socket.Io framework. The application will also have a persistent data storage in order to store course information and login credentials of users, along with additional information like lecture notes. However, it will not store the video data that it provides from students' cameras in any way. To store the user information such as username, password, email address etc., we are planning to use AWS S3 as it is relatively cheap and relatively easy to store and access more complicated documents like PDF.

3.5 Access control and security

Both students and teachers will be able to use our program, which means that multiple actors will have access to different parts of the application. For example,teachers will be able to see the statistics of every person in the class while students can only see their own statistics. Access control is also important in terms of security, since we will store personal information like passwords, so each student must be able to only access their own password. We will have three different actors, teachers, students and administration. Administrators will have no direct access to the sensible user information like passwords, and every student and every teacher will only have access to their own information. Teachers will be able to assign students to the lectures and will be able to receive full lecture statistics, while students will be able to take notes during the class and access those nodes from their profiles.

We will adhere to the General Data Protection Regulation (GDPR) [1]. Following the GDPR regulations, we will encrypt personal data before storing it, making it very hard for others and even administrators to access and read the data. Additionally, after analyzing data from student videos in our servers, we will return the video to the client side and will not in any form store the video contents. If the user presses the "Record" button, the program will ask permission before the recording starts.

3.6 Global software control

Here! will have an event driven control system. It is crucial that multiple users can react with our servers without interruption, as multiple people will be present in a classroom. This can be done using event handlers. Our program will utilize event handlers such that each request will trigger a part of the code asynchronously so that users do not have to wait for other requests to be completed. Whenever a user sends a request, a corresponding subsystem will receive the requests and handle it. The teaching assistant server will constantly receive input from the video feeds of the users. This will send a request from the RTC manager to our Online Teaching Assistant subsystem, which will start the image processing process and gather necessary data. After and while this is happening, a student can save their notes to the database, which triggers warning messages on the client side.

3.7 Boundary conditions

Here! will have three boundary conditions which are initialization of the application, termination of the application and facing failure in the application.

3.7.1 Initialization

The user must have an internet connection and access to a compatible web browser in order to use the program since the program will retrieve and process the data in realtime. For the processing of users videos, and also for the communication between users, connection with the teaching assistant server must be established. The user needs an account registered to the user database in order to use the features of the program.

3.7.2 Termination

The user can log out from the program by clicking the logout button. When the user logs out from the program or closes the browser, any running process in the server created by the user's run will be terminated. If the user intentionally closes the browser without logging out, they will be automatically logged in when they open the website again until the tokens are expired. However, they will be disconnected from any ongoing lecture.

3.7.3 Failure

If the students' internet connection is lost, they will not have access to the server thus the program will not run and they will be disconnected from the lecture. If the browser crashes unexpectedly due to excessive memory usage or browser-related problems, users' connection to the program will be lost but they will be kept logged in until the next time they login by using cookies. If any of the APIs that the program uses become out-dated, some of the features might become unavailable. Database errors might also cause failure of the program and may result in not being able to login and unavailability of saved notes temporarily or permanently depending on the error type.

4 sSubsystem services

In this section, we will explain the major two components of our system: Client and Server.

4.1 Client

Client has two subsystems: View subsystem and Controller subsystem. The View subsystem will be responsible for interface operations and the Controller subsystem will be responsible for the connection between client and server.

4.1.1 View Subsystem

View subsystem handles interaction between the user and our program. This subsystem consists of UI components such as toolbar, menus, pages, pop ups, etc. Users requests and actions are transferred to the Controller subsystem via View subsystem.
View subsystem consists of the following UI components:

Conference Page

This view is responsible for displaying the conference page which consists of camera views of the conference participants, TA notifications, slide view, shared screen view and options to open/close camera and microphone, take note, chat, change settings etc.

Home Page

This view is responsible for displaying an initial home page which consists of the users weekly conference schedules, upcoming conference notifications and options for creating and joining a conference.

Profile Page

This view is responsible for displaying a user's profile page which consists of users information, users' saved notes from previous lectures or list of courses with an "assign student" button depending on the user's role.

Statistics Page

This view is responsible for displaying statistics at the end of a lecture.

4.1.2 Controller Subsystem

Controller subsystem handles users requests and actions transferred from the View subsystem. It is mainly responsible for the connection between the View subsystem and server; fetching data from database, sending image frames collected from users camera to the server, receiving data collected from processed images etc. Controller subsystem consists of Video Manager, Chat Manager and Slide Manager.

Video Manager

Video manager is responsible for the real time flow of the users' videos. It sends the image frames from users' cameras into the server.

Chat Manager

Chat manager is responsible for transferring the messages of in-class chat. It collects the messages sent by the students and the instructor, and sends them to all participants.

Slide Manager

Slide manager is responsible for receiving the slide from the instructor and providing it to the students. Instructors can upload the slide to the slide manager at any time, and students can see the page of their choice even if the instructor is on some other page. However, instructors may force the students to open a particular slide.

4.2 Server

Server interacts with the client in a request-response manner. This is where the requests of the users are handled and responses to these requests are generated. Server consists of Online Assistant Subsystem and Database Subsystem. Online Assistant Subsystem is responsible for processing image frames and other data received from Client. Database Subsystem is responsible for managing the database where all the persistent objects are stored.

4.2.1 Online Assistant Subsystem

Online Assistant Subsystem is responsible for making predictions about the attention of a student towards the lecture from the camera feed frames it receives. This subsystem consists of two major parts: Image Analyzer and Statistics Manager.

Image Analyzer

Image Analyzer is responsible for performing eye tracking, head pose estimation and student-object interactions in order to gather information about a student's attention towards the lecture. It performs these predictions on the frames it receives from the Video Manager. The results of these predictions are then sent to the conference view to be displayed as a TA notification.

Statistics Manager

Statistics Manager is responsible for collecting the analysis results from the Image Analyzer and sending them to the Statistics Page in the View subsystem. Before sending them to the view subsystem, it processes these results into meaningful metrics, to be later viewed on the interface when the conference ends.

4.2.2 Database Subsystem

Database subsystem manages the interactions between system and database. It will communicate with other subsystems in the Server and also with the Controller subsystem of the Client in order to service the requested data. We have 5 to 6 tables in our database that have relationships with each other.

Authentication Manager

One of the tables in our database is the user table which holds information about users such as their login credentials and role. We have seperate tables for the roles of the users, student and instructor. Another table is the course table, which holds information about the courses and there is a relation table which holds information about the relationship between user and course. The Authentication Manager is responsible for handling verification of users actions, register and login requests of the user by fetching data from this user, course and relation tables.

Notes Manager

In order to store notes, we will use S3 services which can easily store PDF images. We will then store the file path in Amazon RDS in order to be able to access the PDFs easily.

5 Consideration of Various Factors in Engineering Design

Public Health

Public health is the motivation of our project. Due to ongoing pandemic, it is crucial that the students remain in their homes, as being in a real classroom is a huge health hazard. Our project ensures that the students remain in isolation while trying to provide an education tool that resembles the feeling of an actual classroom.

Public Safety

Due to the ongoing pandemic, it is essential, regarding public safety, to stay isolated. Due to this factor, our team cannot get together in the same environment as often as we would like, which is making the development process harder in terms of cooperation and teamwork.

Public Welfare

A country’s welfare state affects its average internet connection speeds and accessibility of better electronic devices. Both affect user experience quality of our program, as our program relies on visual data communication, which can be an expensive process regarding internet connection. On the other hand, currently our program is free to use, so it has no real effect on public welfare.

Technological Factors

Technology is constantly developing, and this is an important thing to consider for our project. We need to keep track of this development and implement it to our project so that our work is not obsolete. We also need to look out for similar products, and develop our application accordingly so that our product can add something to the table.

Cultural Factors

Clothings and accessories may differ from culture to culture. These differences affect motion tracking and recognition of facial expressions.

Social Factors

Users need to interact with the program using a language they are familiar with. In order to make our program available world-wide, we need to consider translating our program to different languages. Our program will be English as default, and language might be an issue since approximately 20% of the Earth’s population can speak English.

Global Factors

Global factors do not have a significant impact on the design or analysis parts of the project.

Environmental Factors

Environmental factors do not have a significant impact on the design or analysis parts of the project.

	Effect level	Effect
Public health	10	Public health crisis motivating our project idea
Public safety	5	Hindering the cooperation and teamwork aspect of development due to isolation.
Public welfare	5	Change in user experience quality based on purchasing power
Technological factors	10	Change in design due to support for technologies we use and existence of similar applications
Cultural factors	4	Change in accuracy of image processing because of cultural differences on appearance
Social factors	6	Language barriers between users and the program
Global factors	0	-
Environmental factors	0	-

Table 1: Factors that can affect analysis and design

6 Teamwork Details

6.1 Contributing and functioning effectively on the team

Group members shared the work in a way such that each member is capable of what they are responsible for in order to help them function effectively. This way of sharing minimizes bottlenecks and prevents delaying of the project. It also affects the psychological state of members and keeps them motivated by avoiding feeling of failure.

6.2 Helping creating a collaborative and inclusive environment

Members of the group helped to create an inclusive environment by using communication mediums to discuss and share the work that they create. We arranged meetings while working on reports, introducing new features and debating on problems. During the project we also used Github to make it easier to collaborate on the project. Using Github makes it easier for us to save our work, share and collaborate with others.

6.3 Taking lead role and sharing leadership on the team

Work is divided into work packages and each package is clearly explained in the Analysis Report. Leadership roles are distributed fairly among members to balance the workload as well as allowing them to improve their leadership skills. Leaders of work packages are leading their team by drawing the road map to the completion of the work package and making final decisions on discussions.

7 Glossary

- **Socketio:** Real-time application framework (Node.JS server).
- **WebRTC:** An open framework for the web that enables Real-Time Communications (RTC) capabilities in the browser.
- **AWS:** Amazon Web Services, a package of web services including EC2 and RDS.
- **EC2:** Elastic Computing, Amazon based cloud server.
- **RDS:** Relational Database Services, Amazon based cloud relational database service.
- **S3:** Simple Storage Service, Amazon based cloud data storage service.

8 References

[1] "General data privacy regulation." https://eugdpr.org/. [Accessed: 26-Dec-2019].