



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EVOLI Sync

A video-annotation tool to support the live lesson

TESI DI LAUREA MAGISTRALE IN
COMPUTER SCIENCE AND ENGINEERING
INGEGNERIA INFORMATICA

Author: **Giacomo Cassano**

Student ID: 945642

Advisor: Nicoletta di Blas

Academic Year: 2020-21

Abstract

Starting from the digital age we are living in, the need for new teaching models has arisen. One of these is the so-called HyFlex (i.e. Hybrid and Flexible) model, which is an innovative teaching format that combines (1) face-to-face instruction with online teaching, both (2) synchronous and (3) asynchronous. The central principle of this model is that in all these situations learning must be equivalent, regardless of how/when it is delivered. It therefore turns out to be a model in which the student has the flexibility to move through all three modes when she needs it or when she wants it.

EVOLI Sync is a web platform that supports the HyFlex model, by offering a system based on student-teacher feedback that facilitates the expression of questions and doubts “live”, at a given moment of the lesson, or at a later stage (asynchronously), providing the teacher with an at a glance view of how the lesson is going/went.

The pandemic, which in fact “imposed” the activation of the “hybrid/flexible” class, consisting of students partly in attendance and partly remotely participating in the same lesson synchronously or “later on” (through the access to recorded lessons), has made this model extremely current.

This approach is fully integrated by EVOLI Sync, which therefore supports different types of users: face-to-face students will have a tool to facilitate interaction with the teacher and understanding of the didactic content, remote students will be able to use a more complete teaching experience by benefiting from an innovative digital feedback system to communicate with the teacher and quickly express their level of understanding, while the teacher will be able to monitor in real time the progress of students’ understanding through a feedback analysis and have a set of functions, normally usable in different environments, in a single ecosystem. Moreover, students have the opportunity to collaborate by inserting notes shared between them and tagged on the slides in order to create a handout. The lesson, once finished, is made available to students who will be able to review it asynchronously and, in addition, compare their learning with the feedback given by the audience live.

The tool has been developed as “secular” platform, disconnected from any proprietary system from which, moreover, it could be integrated. In order to be able to demonstrate the efficacy of the approach (in the view of further integrations)

and to allow a quick access to any user, an individual can register and have a tool immediately ready for use. Consequently, EVOLI Sync can also be used in a non-didactic context, in which any person wants to discuss a topic with others and wants to take advantage of a feedback system to understand at first sight what is the reaction of the audience.

The platform has the further objective of disseminating educational content through open courses: the lessons can be made available to anyone and visible in a special section of the platform in order to provide a wide range of courses covering different disciplines.

Recognizing the value and importance of relationship, comparison and collaboration, EVOLI Sync has the ultimate goal of maintaining the benefits of digital technologies to innovate post-Covid teaching.

Key-words: EVOLI Sync, teaching, HyFlex model, feedback.

Abstract in italiano

A partire dall'era digitale che stiamo vivendo, è nata la necessità di avere nuovi modelli didattici. Uno di questi è il cosiddetto modello HyFlex (da Hybrid e Flexible), ovvero un formato di didattica innovativa che combina un approccio istruttivo (1) faccia-a-faccia con l'insegnamento online, sia (2) sincrono che asincrono. Il principio centrale di questo modello è che l'apprendimento risulti equivalente, a prescindere di come/quando viene erogato. Risulta quindi essere un modello in cui lo studente ha la flessibilità di muoversi attraverso tutte e tre le modalità quando ne ha bisogno o quando lo desidera.

EVOLI Sync è una piattaforma web che supporta il modello HyFlex offrendo un sistema basato sui feedback studente-docente che facilita l'espressione di domande e dubbi "live", a un dato momento della lezione, o ad uno stadio successivo (asincronamente), offrendo al docente un'analisi a prima vista di come sta andando/è andata la lezione.

La pandemia, che ha di fatto "imposto" l'attivazione della classe "ibrida/flessibile", costituita da studenti in parte in presenza e in parte a distanza che partecipano alla medesima lezione sincronamente o "on-demand" (attraverso l'accesso a una lezione registrata), ha reso questo modello estremamente attuale

Questo approccio è integrato totalmente da EVOLI Sync, che quindi supporta diverse tipologie di utenti: gli studenti in presenza avranno a disposizione uno strumento per facilitare l'interazione con il docente e la comprensione del contenuto didattico, gli studenti a distanza saranno in grado di usufruire di un'esperienza didattica più completa beneficiando di un innovativo sistema di feedback digitale per comunicare con il docente e manifestare in maniera rapida il proprio livello di comprensione, mentre l'insegnante potrà monitorare in tempo reale l'andamento della comprensione degli studenti attraverso un'analisi dei feedback e disporre di un insieme di funzionalità, normalmente fruibili in ambienti differenti, in un unico ecosistema. In particolare, gli studenti hanno la possibilità di collaborare inserendo degli appunti condivisi tra loro e taggati alle slide in modo tale da creare una dispensa. La lezione, una volta terminata, è resa disponibile agli studenti che potranno rivederla asincronamente e, in aggiunta, confrontare il proprio apprendimento con i feedback dati dalla platea in diretta.

Lo strumento è stato sviluppato come piattaforma "laica", slegata da qualsiasi sistema proprietario nei quali, peraltro, potrebbe essere integrata. In modo tale da dimostrare l'efficacia di questo approccio (in vista di integrazioni future) e permettere un accesso rapido a qualsiasi utente, un individuo può registrarsi ed

avere uno strumento subito pronto all'uso. Di conseguenza, EVOLI Sync può essere usata anche in un contesto non prettamente didattico, in cui qualsiasi persona vuole discutere un argomento con altri e vuole usufruire di un sistema di feedback per comprendere a prima vista quale è la reazione della sua platea.

La piattaforma si pone come ulteriore obiettivo la divulgazione del contenuto didattico tramite corsi aperti: le lezioni possono essere rese disponibili a chiunque e visibili in un'apposita sezione della piattaforma in modo tale da fornire un'ampia gamma di corsi che coprono differenti discipline.

Riconosciuto il valore e l'importanza della relazione, del confronto e della collaborazione, EVOLI Sync si pone come obiettivo ultimo il mantenere i benefici delle tecnologie digitali per innovare la didattica post-Covid.

Parole chiave: EVOLI Sync, insegnamento, modello HyFlex, feedback.

Contents

| | |
|---|-----------|
| Abstract | 1 |
| Abstract in italiano | 3 |
| Contents | 5 |
| Introduction | 8 |
| 1. State of the art | 11 |
| 1.1. Teacher-students interaction in online contexts..... | 11 |
| 1.2. The HyFlex model | 12 |
| 1.3. Video annotation systems | 13 |
| 1.3.1. EVOLI | 13 |
| 1.3.2. Perusall | 15 |
| 1.3.3. VideoAnt | 15 |
| 1.3.4. Democratic Reflection | 16 |
| 1.3.5. EdPuzzle | 16 |
| 1.3.6. VideoNot.es | 17 |
| 1.3.7. Annotate.tv | 17 |
| 1.4. Web conferencing tool used in education..... | 18 |
| 1.4.1. Engageli | 20 |
| 1.5. Summing up | 20 |
| 2. Goal & Requirements | 21 |
| 2.1. Objectives..... | 21 |
| 2.2. Scope..... | 22 |
| 2.3. Scenarios | 22 |
| 2.4. Product modules..... | 24 |
| 2.4.1. Organization of courses and lessons | 24 |
| 2.4.2. Live lesson | 24 |
| 2.4.3. Lesson report | 25 |
| 2.4.4. Open lessons | 25 |

| | | |
|-----------|---------------------------------------|-----------|
| 2.5. | Users | 25 |
| 2.6. | Entities..... | 25 |
| 2.6.1. | State diagrams | 26 |
| 2.7. | Functional requirements..... | 27 |
| 2.7.1. | Authentication | 27 |
| 2.7.2. | Profile | 28 |
| 2.7.3. | Live lesson | 29 |
| 2.7.4. | Report | 30 |
| 2.8. | Non-functional requirements | 31 |
| 2.9. | Domain assumption | 32 |
| 3. | Design & Implementation | 33 |
| 3.1. | Physical application structure | 33 |
| 3.1.1. | 3 level application | 33 |
| 3.1.2. | Benefits of a 3-tier app architecture | 34 |
| 3.2. | Logical application structure | 34 |
| 3.2.1. | MVC pattern | 35 |
| 3.3. | Components and technologies | 36 |
| 3.4. | Data tier..... | 37 |
| 3.4.1. | Conceptual model | 38 |
| 3.4.2. | Logical mapping | 40 |
| 3.4.3. | Data Access Layer | 41 |
| 3.5. | Application Tier | 42 |
| 3.5.1. | Application structure | 42 |
| 3.5.2. | Services | 42 |
| 3.5.3. | API | 46 |
| 3.5.4. | Authentication | 50 |
| 3.5.5. | Socket communication | 51 |
| 3.6. | Presentation tier | 55 |
| 3.6.1. | React application | 55 |

| | | |
|------------------------|---------------------------------------|------------|
| 3.6.2. | Routes and pages | 55 |
| 3.6.3. | Site Map | 56 |
| 3.6.4. | Mockups | 58 |
| 3.6.5. | Components | 69 |
| 3.6.6. | Application global state | 78 |
| 4. | Evaluation | 82 |
| 4.1. | Evaluation context..... | 82 |
| 4.2. | Questionnaire..... | 82 |
| 4.3. | Analysis..... | 92 |
| 5. | Conclusion & Future Works | 94 |
| Bibliography | Errore. Il segnalibro non è definito. | |
| List of Figures | | 99 |
| List of Tables | | 102 |
| Acknowledgments | | 103 |

Introduction

“Think out of the box and create a learning experience where the learner can interact with the content and their brains.” - Rosalie Ledda Valdez

In recent years, the way of teaching has profoundly changed, also due to the need to evolve brought about by the pandemic. Thanks to continuous technological developments, when teaching was no longer able to continue in person, schools and universities were able to continue teaching with the help of numerous teleconferencing applications despite none of these were born with a didactic orientation. In particular, in these platforms there is no system focused on the exchanging of feedback between students and teachers: it is practically impossible for a professor to know the progress of the lesson in terms of understanding students in real time. Very often, in fact, especially in large university classes, the lesson is reduced to the teacher explaining in front of the screen to students with webcam and microphone deactivated. In this case, the distance is felt even more: the context does not allow the student to express his level of learning in a simple and immediate way and the teacher to have a clear idea of trend of the lesson.

What if we could bridge the gap?

This thesis arises among the objectives of proposing an answer to this question. The design and development of a web platform, EVOLI Sync, is then proposed, which creates a video-lesson annotation system capable of supporting the live lesson, offering an innovative infrastructure based on feedback to facilitate interaction between students and teacher and highlight the doubts of the participants at a particular moment of the lesson, whether it is live or not. EVOLI Sync builds, around the live lesson, an interface level between teacher and students that allows the latter to express their level of understanding quickly and intuitively, providing the teacher with a readable analysis at first sight of the progress of the lesson .

In this way, although in distance learning the teacher cannot see the eyes of the students to know if they have understood, he is nevertheless able to know precisely the level of learning of his audience.

It is important to clarify that EVOLI Sync is not exclusive for distance learning, but it is also a useful tool for face-to-face lessons by providing a series of features to record students' doubts, compare with the degree of understanding of other participants. .

Thanks to this approach, the platform aims to support the HyFlex model which indicates a combination of (HYbrid) and flexible (FLEXible) modes. This model harmonizes face-to-face and online teaching, in which each class and learning activity is offered both in person, both synchronous online and asynchronous online. Students have the flexibility to decide how to participate.

During the design of the product, **the goal is not creating a video-conferencing and streaming system but YouTube is used as a support and, in particular, its live broadcasting system, to then build on the feedback-based layer. The overall system we are going to create is classified as a video-annotation tool, that can be proposed and integrated as plugin in the common video-conferencing tools.** We emphasize that we do not want to introduce complexity or distractions in the development of the learning process, but it will be part of the implementation purpose to keep a User Experience as intuitive and simple as possible in such a way as to always keep the focus on the didactic content.

EVOLI Sync supports the pre-existing EVOLI, a tool created to support the flipped-classroom approach, in which the lesson is pre-recorded and viewed by students as a homework assignment, to then discuss the contents and doubts in presence with the teacher. In EVOLI students are allowed to give feedback while watching the pre-recorded video and teachers to have a vision of the students' reaction, knowing which parts are more or less clear and the questions students want to ask, so as to maximize the effectiveness of the face-to-face lesson. The two tools are therefore complementary and together they create a digital context strongly oriented towards teaching.

The aforementioned thesis document is intended to show how EVOLI Sync was first conceptualized and then implemented, showing all its requirements and subsequently its design.

CHAPTER 1, the state of the art, shows the starting context in which EVOLI Sync is placed and explains where the need for an application that meets the platform's requirements comes from. In particular, the HyFlex teaching model is explained and how EVOLI Sync fully supports it is analyzed. Furthermore, differences and similarities are analyzed with similar tools, such as EVOLI and Engageli.

CHAPTER 2 details the objectives and requirements of the platform, showing what users need to be able to do as teachers and students. All the application components

of the product are then explained before analyzing each application requirement in detail.

CHAPTER 3 shows the design of the application, intended for a technical audience. In this section, the 3-level structure of the application is analyzed with the technologies with which it is implemented. Each part of the application is examined to show how the requirements are met by the individual application modules.

CHAPTER 4 shows an evaluation of the application used in two lessons of the course Communication & Argumentation of Politecnico di Milano.

CHAPTER 5 shows the conclusion of the work and how the application could be modified or extended in the future.

1. State of the art

In this chapter we intend to show the state of the art of what the market has to offer at the moment regarding the systems currently used for remote teaching and video annotation systems (especially for education).

In general, the concept of “video annotation” has the meaning of a labelling process used to train AI to detect object and audio in videos. Autonomous vehicles are a notable use case of video annotation. They rely on sensor and camera captured video data that is then labeled with car, pedestrian and other object to train the car where it needs to go, how to react to its environment and what to avoid. Historically video annotators had to synchronize data and tracked objects between frames manually, today AI powered tools are available to help automate tracking and make the annotation process more efficient. Instead, in our thesis this concept is simplified in the sense that the users manually annotate the video lesson with comments, understanding reactions and notes. Each annotation is tagged to the video frame, and a user can seek back to the video instant by clicking on the annotation.

Our analysis begins the description of the teaching theoretical approach we support in EVOLI Sync: the HyFlex model. Then, we will see which video annotation tools are available in the current market. At the end, we will see which platforms are used to supporting the live teaching and which limitations they have.

1.1. Teacher-students interaction in online contexts

“Learning begins with student engagement” (Shulman L.S., 2002)

Classrooms and teacher students relationships can be conceptualized as complex social systems. There are many studies that demonstrate how the interactions between students and teachers are fundamental to understanding student engagement. These relationships can be stabilized by providing teachers knowledge about developmental processes relevant for classroom interaction and personalized feedback/support about their interactive behaviors. It is proved that providing the right supports to teachers’ interactions, student engagement increases. (Pianta et al., 2012). Numerous studies has proved that increasing the number of functionalities of a digital learning environments increases effective engagement of students between the beginning and the end of the lectures. (Hutain & Michinov, 2022).

Starting from 2019, when COVID-19 pandemic had impacted in the world, the education has been influenced at all levels. Institutions and teacher had to quickly respond to an unexpected and forced transition from face-to-face to remote teaching. (Carrillo, & Flores, 2020). This transition to online mode has negatively distorted the way students interact with each other and with the teacher. Often the lessons are carried out through video conferencing software in which each participant has the possibility to interact only by activating the microphone or webcam. Currently, in fact, teaching in university (especially) is translated into transmissive form, which it is expressed through lectures of the frontal type. Furthermore, the fact that university classes are composed by a large number of students (Johnson et al., 2016) which makes it more difficult to arouse interaction and active participation among students. As it is known, instead, the interaction between teacher and students is fundamental, because it is precisely through the teacher-student feedback and vice versa, that you can monitor the processes and improve learning outcomes (Hattie, 2009; Hattie & Yates, 2013).

The interaction therefore becomes extremely poor and, in particular, the teacher is unable to understand if the students are learning the content of the lesson, if not explicitly indicated. Many times, in fact, if a student does not understand the content, he is not even able to express her doubts and the teacher interprets her silence as a false confirmation of learning. This thesis aims to reduce, as far as possible, the distance in online learning processes and improve the interaction between student and teacher by providing participants with controls to implicitly or explicitly report their degree of understanding of the content. Thanks to this data, the teacher is able to observe dashboards to monitor the understanding trend, in order to know if the students are in difficulty or not.

1.2. The HyFlex model

Now we are going to start with the explanation of our theoretical background, that is the HyFlex model, an educational model which is the center on which EVOLI Sync platform puts its focus.

The hybrid flexible, or HyFlex, course format is an innovative teaching format that combines a face-to-face instructional approach with online teaching, where every lesson and activity is delivered in person, synchronously online and asynchronously online. The central principle of this model is that learning is equivalent, regardless of how it is delivered. It therefore turns out to be a model in which the student has the flexibility to move through all three modes when she needs it or when she wants it.

The result is multi-modal learning experience. This approach provides autonomy and flexibility to the students, since they can chose how the modality to join the course.

The central principle to this model is the equality between modalities: all students must have the same learning resources, independently from the way they access to them. Effective use of classroom strategies and/or technology is vital so that all participants can hear verbal interactions.

HyFlex courses are very robust in terms of emergency situations, indeed this approach permits to save and continue the courses in all the situations, such that climate changes, natural disasters, earthquakes, wildfires, hurricanes or health crises (COVID-19), in which there is the need to suspend the F2F modality. (Milman et al., 2020)

1.3. Video annotation systems

EVOLI Sync is defined as a video annotation system. In fact, it build a system of annotation tags on the videos: the feedbacks are labeled to the lesson video instants, and each feedback (comment, reaction, note, ...) contains an anchor to go back to the correspondent video moment in which it is inserted.

EVOLI Sync has a sister platform for supporting the flipped classroom approach, and therefore an asynchronous lesson, Let's now see the similarities and differences between EVOLI and EVOLI Sync.

1.3.1. EVOLI

EVOLI¹ is a tool developed since 2018 to enhance the use of educational videos in courses online. It permits to the students to give immediate feedback and to comment directly the videos during its vision.

EVOLI is a tool developed in a European project (ELSE) which, despite it was released as beta version, the tool has found a considerable use, both in Italy and abroad.

Thanks to EVOLI the teacher can easily point out the parts that are not clear to the students, given that comments and reactions are tagged to the video in the moment in which they are given. The students can stop the video or rewatch it without losing the comments and their evaluations. The overall evaluation of the video is resume in a graph shown to the teacher in the analysis of the lecture.

¹ From <https://www.evoli.polimi.it/>

Currently EVOLI fits into the digital world as tool supporting the flipped classroom approach: the lecture becomes an homework, while the time spent in classroom is used for collaborative activities such as debates or laboratories. In this scenario, the teacher is not the protagonist character but he becomes a sort of facilitator, director of the didactic action.

EVOLI Sync joins EVOLI as its complementary tool, in fact EVOLI puts its focus on the flipped classed approach while EVOLI Sync concentrates on live lectures. Both the platforms offer a feedback-based layer to support the interaction in a classroom between students and instructor. In each of the two tools the feedback are said "tagged", in the sense that each feedback is labelled with the video time in which they are risen up and the tag guarantees the possibility to seek the video (registered in EVOLI and live in EVOLI Sync) to the exact moment in which the feedback is made.

EVOLI has a simpler feedback system with respect to EVOLI Sync. The feedbacks are categorized in reaction and comments: the reaction are two emoji faces, "Get it" and "I don't get it", that can be pressed in a given moment of the video to express the understanding of the user, while the comments are simply strings of text that can be labelled to a video seconds to express a question or to highlight an idea.

The two platforms differ also in the organizational structure: in EVOLI there a clear distinction between the student accounts and the teacher accounts, while in EVOLI Sync each user has the functionalities both as teacher and as student. In fact, each user has its proper student section in which he can enroll himself to some courses and own instructor section in which he can create courses and lectures.

Furthermore, in EVOLI a student can login and watch a video just inserting the video code. In EVOLI Sync a student enrolls to a specific course inserting the course code. EVOLI offers the possibility to login as anonymous. In this way all feedback, reactions and comments, are not associated to a given user. In EVOLI Sync each user must be logged with its own account and its feedback is associated to the user that has made it, but the analysis of the feedback showed in the dashboard during the lecture and in the report does not show to the teacher which students are understanding and which not.

In conclusion, EVOLI Sync is placed side by side with its twin EVOLI in order to fully support the HyFlex approach. All the three educational modalities are implemented inside of the two tools. EVOLI Sync deals with the in-person and synchronous online modality, while EVOLI implements the asynchronous online vision of the lectures mostly applied in a flipped classroom approach, in the sense that the video is seen at home by the students while the time in classroom is spent for content discussion, flipping the normal teaching modality. The effectiveness of

EVOLI Sync is the capability of building, over the HyFlex model, an large feedback-based digital layer between the students following the lecture and the instructor teaching it. This acts as an interface helping the teacher to fully understand the effectivity of his lecture, in particular how much the student, in the different three modalities, have absorbed the content explained.

1.3.2. Perusall

Another video annotation tool for education is Perusall², a free online social annotation platform that enhance the use of the flipped classroom approach. The platform provides an interactive chat to enhance the students engagement while they watch the video incorporated. This technology permit to transform a potentially isolated learning experience to an active, collaborative one in which they can share questions, ideas and receive feedback. In conclusion, the participants can improve their understanding, decrease their boredom and become more comfortable in asking questions. (Sigmon & Bodek, 2022)

Perusall has many similarities with EVOLI, as the primary purpose of the two platforms is to guarantee the support of the flipped didactic model. EVOLI does not provide an interactive chat but creates a useful tool for the teacher to analyze student feedback, while Perusall focuses more on the student side by dedicating to them a communicative space in which to exchange questions, doubts and answers. Both tools create a tagging system, in which chat messages in Perusall and EVOLI feedback are annotated to the video the instant they are posted.

1.3.3. VideoAnt

VideoAnt³ is a web-based software for adding annotations, or comments, to an online video. VideoAnt is a software developed by the College of Education and Human Development at the University of Minnesota. In the platform, a user can insert a new video linking the source (for example a YouTube video) and a new "ant" is created. Viewing the ant, the users can insert all the annotation they want by marking them on the video timeline, in this way the annotations remain tagged to a video instant.

VideoAnt is similar to EVOLI, in the sense that their context is the flipped classroom, in which the annotation are made on a recorded video made available by the teacher.

² From <https://perusall.com/>

³ From <https://ant.umn.edu/>

1.3.4. Democratic Reflection

Democratic Reflection⁴ is a new collective intelligence platform to harness real-time audience feedback over live or video recorded events, such as televised election debates and video testimonials. It is both a new audience interaction method and a new audience feedback gathering and visual analytics technology.

Democratic Reflection is a tool developed by the Open University's Human Centred Computing group (IDea Team) from the Knowledge Media Institute.

It was first designed by an interdisciplinary team of researchers from the Open University and the University of Leeds as a tool for UK citizen to actively engage with political election debates.



In Democratic Reflection the viewers of political debated can express their reactions by using some digital cards. "These cards are reflective statements which try to capture how the debate is engaging viewers as Democratic citizens." (From Democratic Reflection website)

Figure 1.1: Democratic Reflection cards

1.3.5. EdPuzzle

EdPuzzle⁵ is a free video lesson platform developed specifically for education settings. Teachers generate and share a class code with their students to give them access to their video lessons. The editor allows the users to manipulate the video used for your lesson in different ways:

- Crop the video
- Add audio comments or an entire audio narration track.
- Embed images, PDF documents or hyperlinks.

⁴ From <https://democraticreflection.org/>

⁵ From <https://edpuzzle.com/>

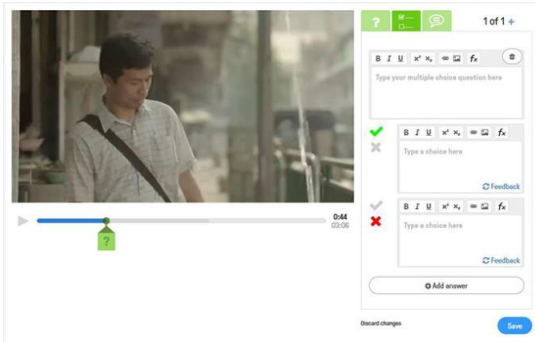


Figure 1.2: EdPuzzle

The platform permits the teacher to insert quizzes or questions in order to engage the students and test their understanding level.

All these features permits to keep the students active as they watch the video, reducing the boredom of staying in front of a screen and enhancing the interaction with the platform. It is demonstrated that adding a sort of gamification to the lesson content permits to increase the attention

and the engagement of the students (Mischel, 2019)

1.3.6. VideoNot.es

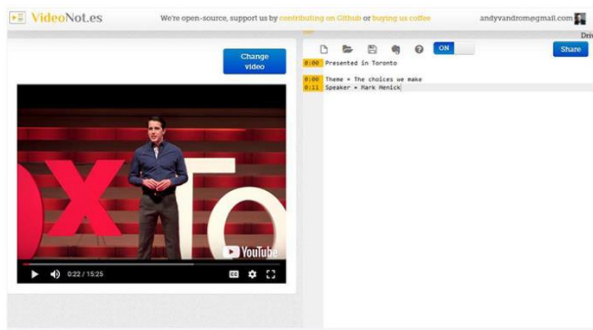


Figure 1.3: VideoNot.es

VideoNot.es⁶ is a video annotation platform that permits students to collaboratively take notes on video content from YouTube, Vimeo and popular MOOC platforms. The video is placed side by side with a text editor, with which students can take notes that are automatically synchronized with the video. Student can click on the notes to seek to the relevant part of a

video, while dragging a marker to a specific timestamp brings up the notes taken on that selection. The notes are shared between participants and this sort of collaboration makes VideoNot.es a powerful platform for students that may want to work together sharing knowledge.

1.3.7. Annotate.tv

Annotate.tv⁷ is a video annotation tool that permits to add time stamped notes on any YouTube video. It works pasting a YouTube link in the interface and then playing the video and add your annotations. Annotations can include links and the notes are exportable in PDF⁸.

⁶ From <https://videonot.es/>

⁷ From <https://annotate.tv/>

⁸ From <https://media-and-learning.eu/type/tools/make-annotations-on-youtube-videos/>

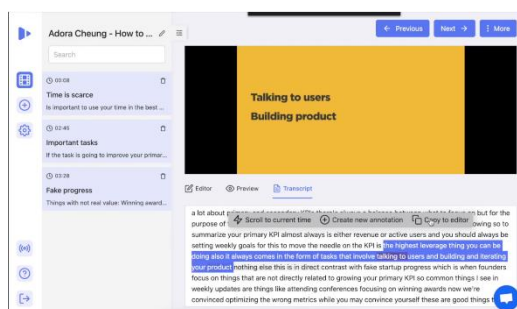


Figure 1.4: Annotate.tv

1.4. Web conferencing tool used in education

Starting from the pandemic Covid-19 years, the concept of distance learning has developed more and more. There was therefore a need to use the classic web conferencing apps such as Microsoft Teams, Zoom, Google Meet with the aim of supporting distance learning. At that moment it was necessary to find a solution quickly and without warning. The tools worked and the teaching was able to continue despite the fact that none of the tools used were oriented towards education.

Nowadays, Zoom is the most popular video conferencing app with over 200 million daily users. Zoom has features like supporting 100 video participants and allowing 49 videos on screen. Also, it supports screen sharing, screen recording, team chats and searchable history.

Microsoft Teams is a direct competitor of Zoom having similar functionalities. Microsoft said that 183,000 education organizations, which include many schools, in 175 countries were now using Microsoft teams for education.

Cisco WebEx Teams is a video conferencing software developed by Cisco. It permits up to 200 video users to join a video meeting. The platform offers important integration features with third party applications like OneDrive, Google Cloud and others.

GoToMeeting is a web platform developed by LogMeIn. It is an online software package which is used for online meetings, sharing desktop screens and video conferencing. Similar to WebEx Teams, GoToMeetings can also integrate with many third-party applications and software which is used for increased productivity in the office and online meetings.

In conclusion, a comparison table follows that compares some of the features of the tools:

Table 1.1: Comparison between web conferencing tool used in education

| Parameters | Zoom | Meet | Teams | Webex | GoToMeetings |
|---------------------------------|--|------------------------------------|----------------------------------|------------------------------|-------------------------------------|
| No. of participants | 50-500, 10000 (webinar) | 25, 100 (G-suite users) | 250-10000 (paid) | 1-3000 | 1-1000 |
| Authentication mechanism | Poor authentication mechanism | Uses SHA-1 HMAC for authentication | Uses OAuth2.0 for authentication | NA | NA |
| Authorization mechanism | No | Yes | Yes | NA | NA |
| Encrypted communication | No | Yes | Yes | 256-bit AES,PKI | Yes |
| Compatibility | IOS, Android, Microsoft, Linux (partial) | IOS, Android, Microsoft, Linux | IOS, Android, Microsoft, Linux | IOS, Android, Windows, MacOS | IOS, Android, Windows, MacOS, Linux |
| Whiteboard | Yes | No | Yes | Yes | Yes |
| Recording | Yes | Yes | Yes (paid) | Computer and services based | Yes |
| Video quality | HD | HD | VGA, HQ | VGA, HQ, HD | VGA, HD |
| Chat | Yes | Yes | Yes | Yes | Yes |
| Screen sharing | Yes | Yes | Yes | Yes | Yes |

(Singh & Awasthi, 2020)

After this comparison it is possible to say that these platforms represent a class of tools that, nowadays, are used for teaching, but none of these are born with this orientation. In particular, these platforms lack a level that places the focus on the interaction between student and teacher making it difficult to understand:

- The level of understanding of the students
- What are the doubts of the students
- Which parts of the lesson are unclear

At this point it is important to clarify that there are not yet many video conferencing tools that place their focus on teaching and that is why EVOLI Sync takes on a very innovative connotation. Now we are going to analyze a platform still under development born with the same purpose as EVOLI Sync, although with some significant differences that we will highlight.

1.4.1. Engageli

Engageli⁹ is digital platform founded by Dan Avida and Daphne Koller (also founder of Coursera) which shares the same idea and objectives of EVOLI Sync and EVOLI, i.e. the fact that, in this current digital era, there exist a lot of video conferencing tools used for learning despite none of these ones have a user experience focused on the didactic process. In other words, students and teachers are seen as users of a classical video platform application but they are not able to support the lecture and the interaction between student and teachers with appropriate tools.

In this tool students can enter in a classroom and join a table, that is a limited group of classmates. Student in a group can use the available tools, such as chat, screen sharing and notes privately. Engageli offers to the students the possibility to interact with chat, notes, question & answers and shared documents. This functionalities are very similar with respect to EVOLI Sync: the chat permits to send messages either to all classmates, the groupmates or to only the instructor, the notes represents (as EVOLI sync) a shared collection of text and images in which the students share their own notes about the content of the lecture, the Q&A section permits to the students to post question and receive answer.

Despite having a lot of similarities, EVOLI Sync puts his focus on making the teacher able to understand whether the students are understanding the content explained through the feedback dashboard that builds a live and “at-a-glance” analysis based on the students’ understanding reactions.

1.5. Summing up

The purpose of this thesis is therefore to create a tool that not only allows you to follow a remote lesson but also provides tools that enrich the teaching experience and place the exchange of feedback between students and teachers at the center of the educational project. The desired result is to have a video-annotation platform that can improve the interaction of both the lessons followed at home and the lessons attended in person and finally can build a report on the progress of the lesson. In other words, we want it to be able to support all modes of the HyFlex model.

⁹ From <https://www.engageli.com/>

2. Goal & Requirements

2.1. Objectives

EVOLI SYNC is a web platform belonging to the category of video-tagging annotation tools for teaching. Its primary objective is to support live teaching, offering a feedback-based system to facilitate understanding and express the presence of doubts on the part of the student at a given moment of the lesson but also to provide the teacher with useful information to evaluate at first sight the progress and level of understanding of the lesson by the participants.

The pandemic, which in fact "imposed" the activation of the "hybrid/flexible" class, consisting of students partly in attendance and partly remotely participating in the same lesson synchronously or "later on" (through the access to recorded lessons), has made this model extremely current. EVOLI Sync is therefore also proposed as a tool capable of assisting the so-called HyFlex model. The platform offers a tag-based feedback system to support student-teacher interaction during lessons of this type. In this way:

1. Distance students will be able to take advantage of a more complete teaching experience by benefiting from an innovative digital feedback system to communicate with the teacher and the rest of the class.
2. In-person students will have a tool that brings together numerous features to facilitate interaction with the teacher and understanding the teaching content.
3. The teacher will be able to monitor in real time the progress of student understanding through a system capable of analyzing the feedback and will have a set of functions to better prepare and manage the lesson, normally usable in different environments, in a single ecosystem.

The platform also tries to implement its own concept of open courses, in which a user can see not only of recorded lessons but also live lessons without being registered to the course. The teacher, in fact, can open his lesson to a larger audience, making it available to a very wide audience.

EVOLI Sync has the ultimate goal of innovating teaching by applying the benefits of the digital world to live lessons by creating a feedback interface to bridge part of the gap in online teaching by adding a high degree of interactivity between students and teacher.

2.2. Scope

EVOLI-Sync proposes itself as a secular platform, that is, disconnected from any school or academic organization. In this way, an individual can register and have a platform available immediately ready for use. It is therefore not necessary to have a company or university account. Consequently, EVOLI Sync can also be used for a non-didactic context, in which any individual wants to discuss a topic with others and wants to take advantage of a feedback system to understand directly what is the "reaction" of his audience.

DEFINITION, ACRONYMS, ABBREVIATIONS

- **EVOLI-Sync:** part to be integrated with EVOLI to support live teaching
- **Video-tagging annotation system:** a system that allows users to interact digitally with a video by associating certain moments of the video with some feedback.
- **Presentation:** slides in PDF format
- **Normal videos:** YouTube videos recorded and already uploaded (not Live)
- **Live:** YouTube Live video
- **Open Lessons:** lessons that can be seen without being registered to the platform.

2.3. Scenarios

This section shows some possible scenarios for using EVOLI-Sync with the aim of demonstrating that the platform can be used in different areas, from the most "official" such as a university lesson, to the simplest such as a group of school repetitions.

SCENARIO 1: UNIVERSITY CLASS

Starting from the years of Covid-19, the concept of mixed class has increasingly developed, i.e. a class that partly follows the lesson in the presence and partly at a distance, especially in the university environment. Professor Rossi of the Politecnico di Milano decides to use EVOLI-Sync for his lesson, in which students with equal freshmen follow remotely and the others in attendance. He creates the course "Mathematics-Rossi" and insert a new lesson. The new lesson is titled "Differential Calculus". The teacher also associates a PDF presentation with the slides of the lesson. Finally, the teacher prepares a live stream on YouTube and associates the live link with the lesson. At this point the lesson is ready and can be followed by students who access the platform, enter the course and follow the live lesson. Through monitoring the feedback, the teacher notes that during the explanation of

the definition of "derivative" the percentage of understanding has dropped below 50%, he then decides to re-explain it until almost all students report an understanding score of 2/2. At the end of the topic, the teacher decides to answer the questions in the chat.

SCENARIO 2: SCHOOL TUTORING

Marco is a university student who teaches English to a group of middle school kids. He decides to organize a meeting on EVOLI-Sync in such a way that he can do tutoring to all the boys at the same time, receiving feedback in real time to see how much they are understanding. He then organizes his lesson on EVOLI-Sync, creating the live on YouTube and inserting the link in the lesson. The students follow the lesson on EVOLI using the feedback system to report how much they are understanding the lesson. In addition, they communicate via chat with Marco.

SCENARIO 3: CLASS DISCUSSION

EVOLI Sync can also be used by a classroom totally in attendance. Class 3 ^ B of the Leonardo da Vinci's high school is taught by Professor Verdi in philosophy. Prof Verdi decides to take an alternative lesson to stimulate the class to discuss the different thoughts of the philosophers covered during the year. Prof Verdi then carries out the lesson on EVOLI Sync and asks to post his thoughts on the philosophical line presented on each slide in a public note to discuss the various philosophies. At the end of the lesson, students download the handouts containing all the thoughts of the class and discuss it during the next lesson with the teacher.

SCENARIO 4: REPORT

Beatrice is a university student of Nursing. She has just finished the lesson about Nursing Methodologies and she had difficulty following the lesson. Once back home she decides to listen to the lesson again through the report on EVOLI Sync. From there he can see the doubts of his teammates from the special question box and thanks to the answers given he is able to solve all his doubts.

SCENARIO 5: OPEN LECTURE

Giacomo is an high school student at the last year that would like to enroll to Politecnico di Milano as university. He would like to follow a university lesson to better understand how university is structured. Giacomo open EVOLI Sync and goes to the open lecture section. He select a live lecture from the open ones and follow it from home.

2.4. Product modules

This section presents the macro-functionalities of EVOLI-Sync. For convenience, the list is divided into categories that represent the different application components that are created.

2.4.1. Organization of courses and lessons

- **AUTHENTICATION:** Users will be offered the opportunity to register. Each user can log in and log out of the platform.
- **PROFILE:** Each user will have a student section and a teacher section. In the student section you can view the courses you are enrolled in, register for new courses and see, for each course, the details of your lessons. In the teacher section the user can create courses and lessons.

2.4.2. Live lesson

During the course of the lesson, several features will be available that we are going to categorize.

VIDEO LESSON

- **LIVE VIDEO:** Participants must be able to view the live video of the lesson. EVOLI Sync will rely on YouTube for the reproduction of live videos.
- **VIDEO TAGGING SYSTEM:** The student must be able to use a tagging-system to insert feedback while watching the live video of the lesson.

SLIDES

- **SLIDES PRESENTATION:** The teacher must be able to share the presentation (slides) to the students, who can view it while they follow the lesson.

FEEDBACK SYSTEM

- **FEEDBACK DASHBOARD:** the teacher must be able to monitor the progress of the lesson quickly and easily through an automatic live analysis of student feedback. The interface is shown, if the teacher allows it, also to students.

COMMUNICATION SYSTEM

- **CHAT:** Participants in the lesson must be able to communicate via chat. The chat can be enabled or disabled by the teacher.
- **QUESTIONS:** The student must be able to enter questions that will be tagged at an exact moment in the video. Lesson participants must be able to

answer questions. Each question can have a status associated with it: open or resolved editable by the question creator.

- **NOTE SYSTEM:** Lesson participants must be able to create notes and tag them, if there is a presentation, to a given slide. Notes can be private to the participant or shared with the rest of the class.
- **REACTIONS:** Lesson participants should be able to express reactions regarding chat messages, memos and answers to questions. The reactions indicate whether the "likes" or "dislikes" element.

2.4.3. Lesson report

- **REPORT:** Once the lesson is finished, ENDED status, the participants must be able to access a report containing: the questions asked during the lesson with the relative answers, all the notes and an analysis of the feedback received. In the report, you can review the lesson and PDF presentation.

2.4.4. Open lessons

- **OPEN LESSON:** The platform offers a section related to open lessons, where the audience can see recorded lectures or attend live lectures that are open to all.

2.5. Users

In EVOLI-Sync, the user registers without specifying whether he is a teacher or a student. In this way, a user can be both a teacher of some courses and a student in other courses. In particular, a user can have a list of courses taught and a list of courses he is enrolled in as a student.

2.6. Entities

In this section we are going to show the entity diagram to identify the various components of the system that we want to create:

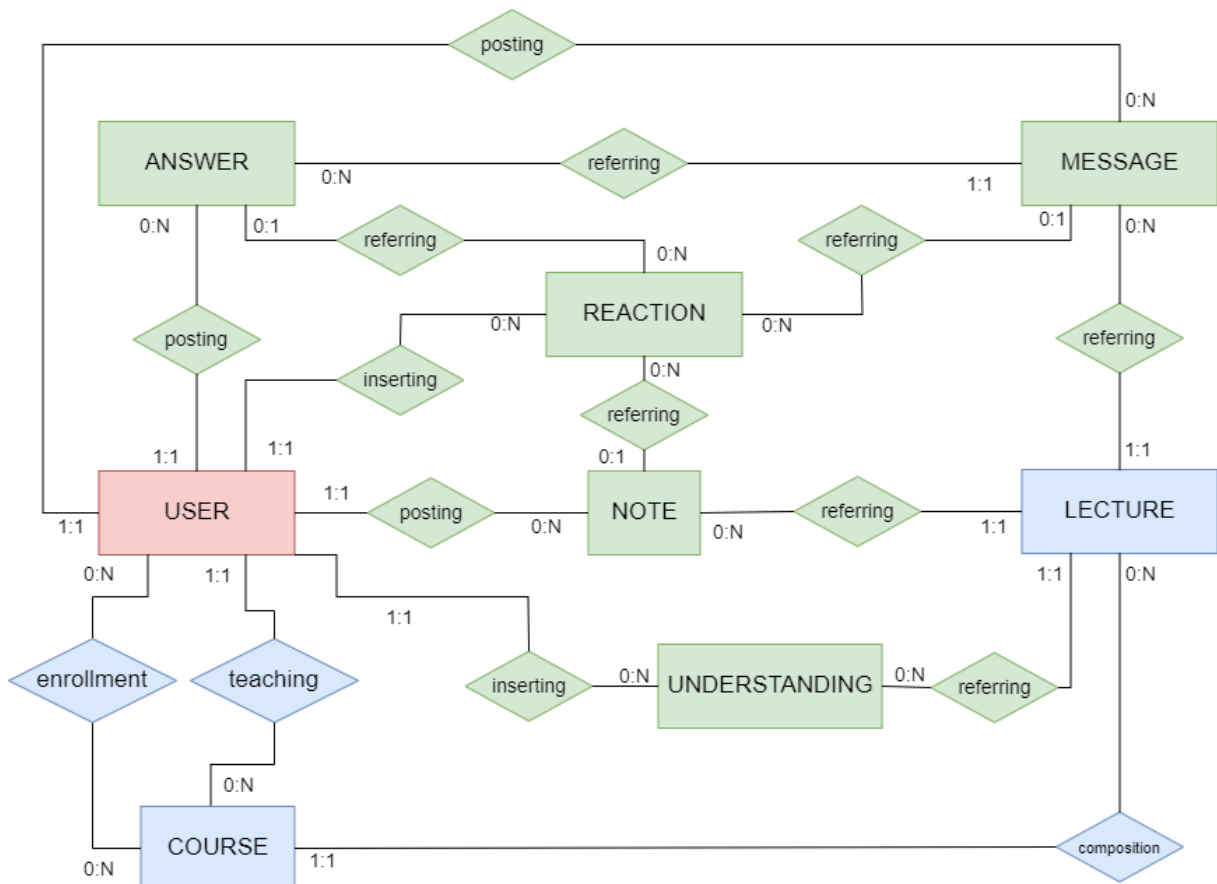


Figure 2.1: Entity Relationship diagram

In particular, we are going to identify:

- The users
- The organizational structure of the lessons: each teacher teaches courses, to which students are enrolled. Each course is made up of lessons, which in turn have their own feedback system.
- The feedback system: We distinguish the chat, the answers to the questions, the notes and the understandings (I understand, I don't understand, I half understand).

2.6.1. State diagrams

This section shows the life cycle of a lesson:

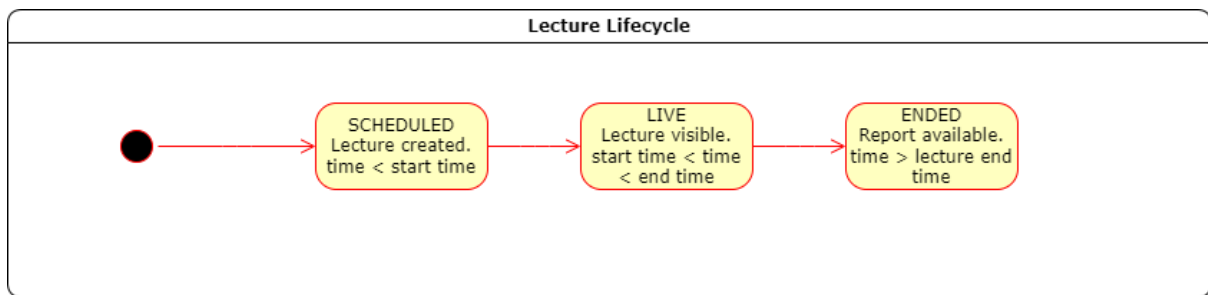


Figure 2.2: Lesson state diagram

A lesson can have 3 distinct states:

- **SCHEDULED**: when the teacher creates a new lesson. In this state, the lesson has been scheduled, i.e. the teacher has entered the necessary data to send the lesson live and make it available to students. The lesson data is still editable.
- **LIVE**: the lesson is live, the current time is between the beginning and the end of the lesson. Students and the teacher can access the live page to follow the lesson.
- **ENDED**: the lesson is over. The teacher and the students can consult the report.

2.7. Functional requirements

This section shows the functional requirements of the EVOLI-Sync system, divided by category.

For each requirement we will indicate :

- **SIDE** : specifying with S if it is a function allowed to a user who follows the lesson as a student or D if it is a function allowed only to the teacher of the lesson.
- **COD** : the identification code of the requirement
- **REQUIREMENT** : the text of the requirement

2.7.1. Authentication

Table 2.1: Authentication requirements

| SIDE | COD | REQUIREMENT |
|------|-------|--|
| SD | AUTH1 | Users must be able to register on the platform by entering email, name, surname, password. |

| | | |
|----|-------|--|
| SD | AUTH2 | Users must be able to log in by entering their email and password. |
| SD | AUTH3 | Users, when logged in, must be able to log out of the platform. |
| SD | AUTH4 | Users must be able to change their profile data: name, surname and password. |

2.7.2. Profile

Table 2.2: Profile requirements

| WHO | COD | REQUIREMENT |
|-----|-------|---|
| D. | ORG1 | A user must be able to view, insert, delete courses from their list of taught courses. |
| D. | ORG2 | A user must be able to view, insert, delete, modify the lessons of the lesson list of a taught course. |
| S. | ORG3 | A user must be able to enroll in a course by entering the course code. |
| D. | ORG4 | A user must be able to get the code of a course he teaches. |
| S. | ORG5 | A user must be able to view or delete courses from their list of courses taken. |
| S. | ORG6 | A user must be able to view a course's list of lessons from their list of courses taken. |
| D. | ORG7 | A user must be able to create a new lesson of a course he teaches by entering: name, start date and time (after the current one), end date and time, URL of the live YT and slide presentation. |
| D. | ORG8 | A user must be able to edit the lesson data. If the lesson is over it is not possible to change the times, presentation and video URL. |
| SD | ORG9 | A user must be able to download the lesson slides, if they exist. |
| SD | ORG10 | A user must be able to download a PDF containing the notes shared by all users plus their private individuals, in order to create a "handout" of the lesson. |

2.7.3. Live lesson

Table 2.3: Live lesson requirements

| WHO | COD | REQUIREMENT |
|-----|--------|---|
| D. | LIVE1 | The teacher must be able to follow the live session of the lesson, viewing the video player, the slide presentation, the feedback dashboard and the communications dashboard on a single screen. |
| DS | LIVE2 | A user must be able to log in and see the live session of the lesson between the scheduled start time and the scheduled end time. |
| DS | LIVE3 | The video players must show the YouTube live stream of the lesson, allowing participants to go back and forth, stop and resume the live stream. |
| D. | LIVE4 | Participants must be able to manage the progress of the presentation slides during the lesson. In particular, the teacher can scroll the slides back and forth, go back to the beginning, go to the end or select a specific page. |
| S. | LIVE5 | Students must be able to scroll through the slides regardless of which slide the teacher is presenting at that particular moment. |
| S. | LIVE6 | Students must be able to return to the slide presented by the teacher at that particular moment. |
| S. | LIVE7 | A user must be able to create notes and tag them to the slide they are viewing . The note can be public or private. If it is private, it is viewable only to the creator. |
| SD | LIVE8 | The user must be able to create notes even when there are no slides. In this case they will not be tagged to any slide. |
| | LIVE9 | The user must be able to see the notes tagged on each slide or, if there are no slides, see all the notes. |
| SD | LIVE10 | Participants of the lesson must be able to exchange messages via chat. Chat messages are differentiated into simple messages and questions. Questions are messages that have a status (OPEN or SOLVED) and a tag at the instant of the video in which they are asked. |

| | | |
|----|--------|--|
| SD | LIVE11 | The creator of a question must be able to change its status from OPEN to SOLVED. |
| SD | LIVE12 | The user must be able to enter answers to questions. |
| SD | LIVE13 | The user must be able to insert / remove LIKE, NOT LIKE reactions on chat messages, answers to questions and memos. |
| S. | LIVE14 | The student must be able to insert an understanding reaction that says on a scale of 0 to 2 how much he is understanding the content of the lesson, where 0 is "I don't understand anything" and 2 is "All clear". At the beginning of the lesson the default user is understanding 2. |
| SD | LIVE15 | The user must be able to view the understanding percentage of the last 5 minutes, the% is calculated like this: $\% = (\# 2 + 0.5 \# 1) / (\# 2 + \# 1 + \# 0) * 100$ where #x is the number of students who are understanding x. |
| SD | LIVE16 | The user must be able to see the number of students he is understanding 0, 1 or 2 at that particular time. |
| D. | LIVE17 | The teacher must be able to view the percentage of understanding up to that particular moment of the lesson. The percentage is calculated as in the LIVE14 requirement. |
| D. | LIVE18 | The teacher must be able to notice through a visual signal if the percentage of understanding falls below 50%. |
| D. | LIVE19 | The teacher must be able to disable chat, notes and the display of the student feedback dashboard at any time during the lesson. |
| D. | LIVE20 | The teacher must be able to finish the lesson. At that point the participants automatically exit the lesson. |

2.7.4. Report

Table 2.4: Report requirements

| WHO | COD | REQUIREMENT |
|-----|------|---|
| SD | REP1 | Users, once the lesson is finished, must be able to access the lesson report. |

| | | |
|----|------|---|
| SD | REP2 | Users must be able, on the report page, to review the lesson. |
| SD | REP3 | Users must be able, on the report page, to view the slides of the lesson. |
| SD | REP4 | Users must be able, on the report page, to view the summary of the questions asked during the lesson, with the relative answers and to insert new ones. |
| SD | REP5 | Users must be able, on the report page, to view the notes entered during the lesson and insert new ones. |
| SD | REP6 | Users must be able, on the report page, to view a graph representing the trend and% understanding of the entire lesson. |
| S. | REP7 | Students must be able to visualize their understanding progress in the lesson, through a color chart where the red areas are the intervals with understanding 1, the yellow areas with understanding 1, the green areas with understanding 2 and the blue areas where the user was not logged in. |
| SD | REP8 | The teacher must be able to disable chat, notes, and viewing the student feedback dashboard at any time. |

2.8. Non-functional requirements

HARDWARE REQUIRED

EVOLI-Sync is a website, therefore accessible from any browser from PC, Tablet or Smartphone.

PORTABILITY

The EVOLI-Sync platform must be available and responsive for each device, in particular PC, smartphone and tablet.

AVAILABILITY

The application must be accessible 24 hours per day.

LANGUAGE

The user must be able to choose whether to view the platform in Italian or English.

2.9. Domain assumption

Each teacher must have a Google profile. This assumption is necessary to allow the teacher to run YouTube live from his own channel.

As already mentioned, EVOLI Sync depends on YouTube for the creation and viewing of live videos. In particular, the application makes use of two APIs: the YouTube IFrame Player API and the YouTube Data API.

YOUTUBE PLAYER API

It allows you to embed a YouTube video player on any website and to control the player via JavaScript. The website version is the IFrame Player API , which inserts the video player into an <iframe> tag on the page. In addition, it identifies the different types of events that the API can send and describes how to write event listeners to respond to them.

EVOLI Sync uses this API to embed video players on different pages.

YOUTUBE DATA API

It allows you to query the YouTube database to extract data about videos, playlists, channels and, in general, incorporate functions that can normally only be performed from the YouTube interface. It allows you to treat various YouTube elements as JavaScript objects.

In our case, the data API allows us to extract the YouTube channels of each individual user and retrieve the id of the video currently streaming on the user's channel.

3. Design & Implementation

The purpose of this chapter is to provide the design description of the EVOLI Sync project. In particular, the application will be detailed under the physical and logical point of view to give a complete view of its architecture. We will then see how the requirements described above are implemented by the individual application components.

In conclusion, the goal of this section is to provide an exhaustive description of each level and application component.

3.1. Physical application structure

3.1.1. 3 level application

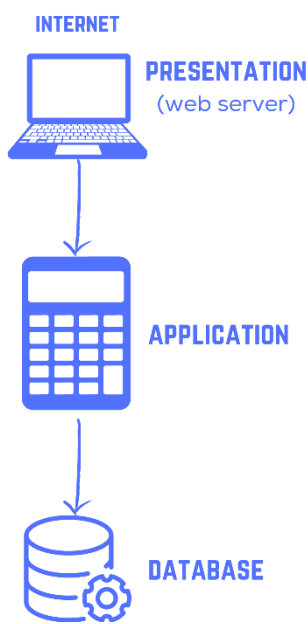


Figure 3.1: 3 level application

EVOLI Sync follows the classical 3 tier application model where “tier” has the meaning of logical layer. A 3-tier application architecture is a modular client-server architecture that consists of a presentation tier, an application tier and a data tier. The data tier stores information, the application tier handles logic and the presentation tier is a graphical user interface (GUI) that communicates with the other two tiers. The three tiers are logical, not physical, and may or may not run on the same physical server.

PRESENTATION TIER

This tier is built with HTML5, cascading style sheets (CSS) and JavaScript. This tier is accessed by a computer device through a web browser and communicates with the application tier through application program interface calls.

APPLICATION TIER

The application tier contains the business logic that implements the application's core functions. It offers some API to exchange data with the presentation tier and in our case has a direct access to the data tier though an Object-Relational Mapping

(ORM), that is a technique that permits the consistency of data and perfectly integrates the entities of a relational database into the object oriented programming.

DATA TIER

The data tier consists of a database and a program for managing the CRUD operations on its data. This tier permits the maintaining and the persistency of the application data. In our casa the data tier consists in a relational database.

3.1.2. Benefits of a 3-tier app architecture

The benefits of using a 3-tier architecture include improved horizontal scalability, performance and availability. With three tiers, each part can be developed concurrently by a different team of programmers coding in different languages from the other tier developers. Because the programming for a tier can be changed or relocated without affecting the other tiers, the 3-tier model makes it easier for an enterprise or software packager to continually evolve an application as new needs and opportunities arise. Existing applications or critical parts can be permanently or temporarily retained and encapsulated within the new tier of which it becomes a component.¹⁰

3.2. Logical application structure

Now let's see in detail what are the logical components that make up each level and how these components communicate with each other. Later we see with what technologies they are implemented.

¹⁰ From <https://searchsoftwarequality.techtarget.com/definition/3-tier-application>

3.2.1. MVC pattern

It is easy to see how the application architecture follows the Model-View-Controller architectural pattern, which separates the architecture into software components that implement 3 fundamental roles¹¹:

- The **model** conceptualizes the various entities involved in the domain
- The **view** allows you to create an interface that shows the model data in an appropriate format and allows the end user to communicate with the application logic displays the data contained in the model and deals with the interaction with the end user
- The **controller** contains all the business logic and changes the state of the model after receiving the appropriate commands from the view.

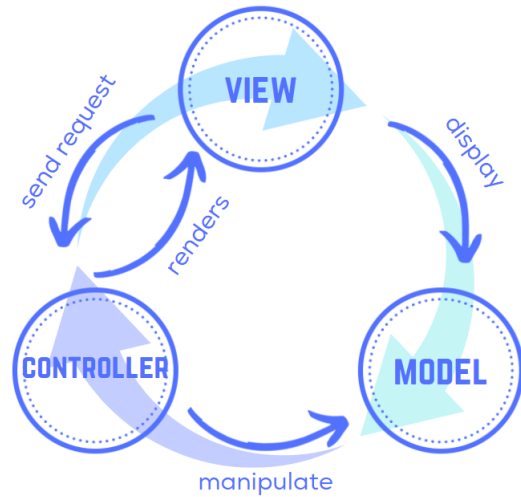


Figure 3.2: MVC pattern

EVOLI Sync, being a web application, maps the view to the client browser, the controller in the application server application logic and the model in the application state.

¹¹ From <https://it.wikipedia.org/wiki/Model-view-controller>

3.3. Components and technologies

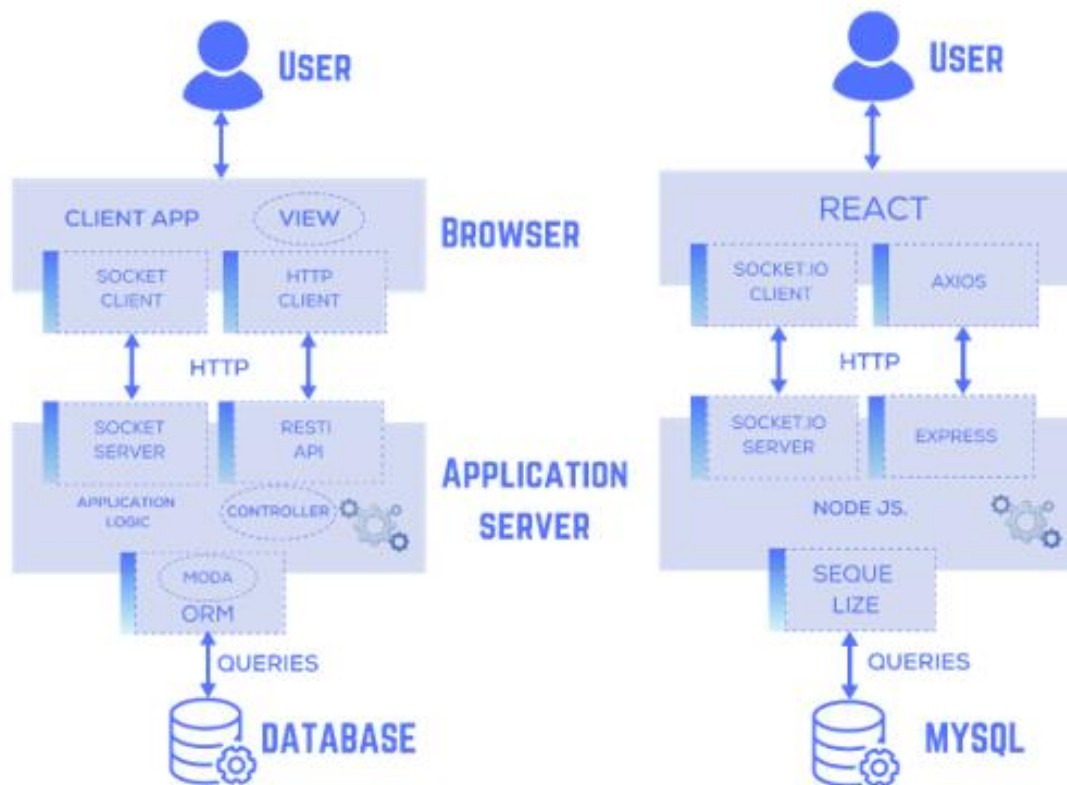


Figure 3.3: Components mapping and technologies

DATABASE: The application database follows a relational model and is implemented with MySQL, a totally open source DBMS.

ORM: The Object- Relational Mapping we will use is Sequelize. An ORM is a programming technique that allows you to map the relational tables of the database with classes and its records to objects belonging to the application logic of the server. Sequelize is a promise- based ORM implemented in Node.js (also) for MySQL.

APPLICATION LOGIC : By application logic we mean all business logic, that is the algorithms that receive events from views and modify the state of the application. It therefore consists of the processing core that makes the application operational. The EVOLI Sync application logic is developed with Node.js, an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.

REST API: The rest API defines a set of constraints of how a distributed architecture on the web should behave. In particular, it provides independent, uniform and reusable interfaces to facilitate the exchange of information between client and server. These interfaces are accessible via http methods and are allocated on specific

URLs and allow the interchange of data in standard formats, in our case JSON. EVOLI Sync relies on Express as a flexible and lightweight framework for building APIs.

WEB SOCKETS: WebSocket is a web technology that provides full-duplex communication channels through a single TCP connection. In our case we will use Socket.IO, a JavaScript library for real-time, event-oriented web applications. In our application it will be especially useful to manage the exchange of feedback visible in real time during the live viewing of the lesson.

HTTP CLIENT: The http client is the component responsible for making calls to the REST API present on the server. In particular, it deals with making http requests and receiving its replies formatted, in our case, in JSON. We will use axios as the http client.

CLIENT APPLICATION : The client application in question is a web application that can be viewed from any web browser. It is built with React, a JavaScript library for creating user interfaces comparable in functionality and application power to the Vue.js and Angular frameworks.

3.4. Data tier

In this section we are going to see how the entity relationship diagram in the requirements chapter is translated into a logical diagram. It is therefore now possible to see the final database schema with the attributes of each single table.

In conclusion, let's see how each entity is mapped by Sequelize on a specific class of the model.

1.1.1. Conceptual model

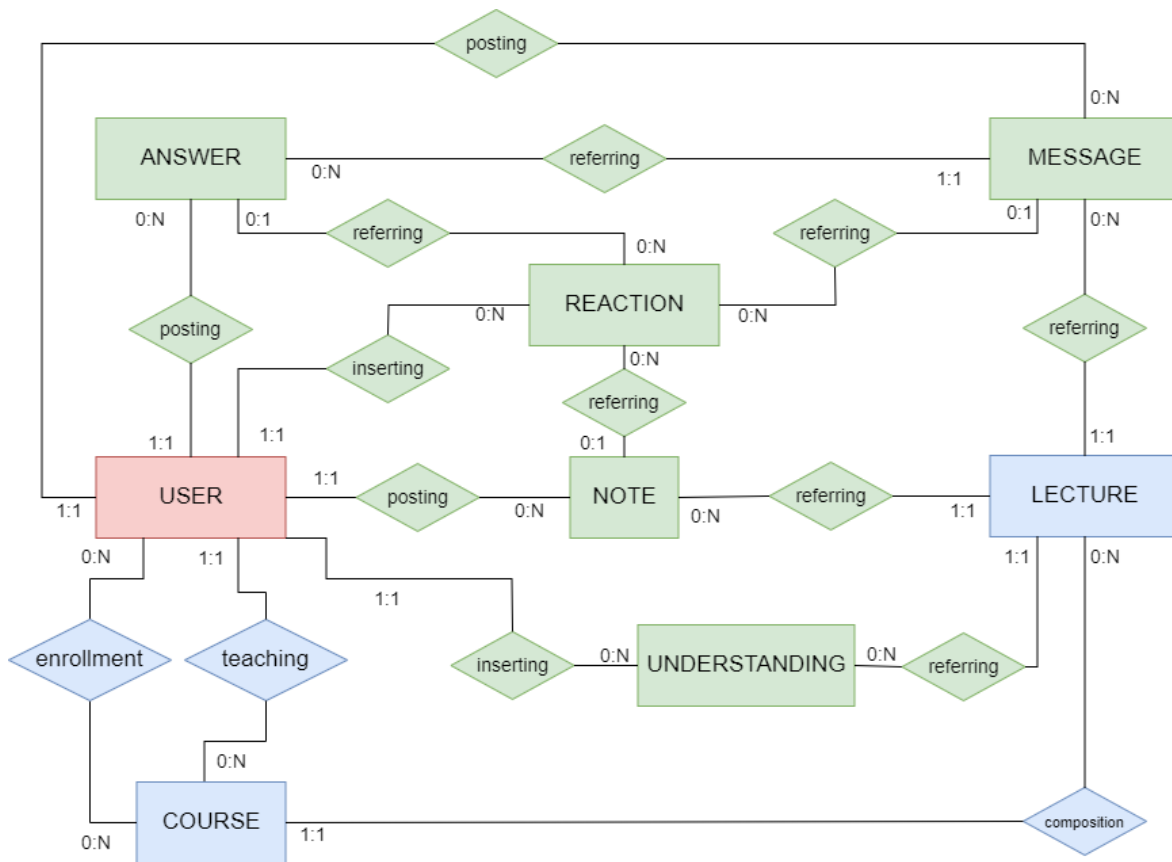


Figure 3.4: ER diagram

- **USER:** each user is identified by his email, during the registration phase his name, surname and password are provided. The password hash is saved in the database . The pair <email, password> constitutes the credentials to be provided upon access.
- **COURSE:** each course is identified by a unique id and a UUID. It has a name and a single teacher.
- **ENROLLMENT:** represents the enrollments of studies in courses. The enrollment is identified by the pairing <student email, course id>. Currently there is an attribute representing the status for scalability reasons: if you wanted to make some courses on request, when the student requests registration, the status would be 0, when the teacher accepts the request it would be 1 or in the case refused it would be 2.
- **LECTURE:** each lesson is identified by a unique id and a UUID. It has a name, a start date, an end date, the link to the YouTube video and, if entered, the URL of the PDF presentation. Furthermore, the enabling, decided by the teacher, of the functionality is recorded: chat, notes and viewing of feedback

from students. Each lesson belongs to a course. The lesson can be made available as a MOOC.

- **UNDERSTANDING**: represents the learning reaction. It is identified by a unique id and a UUID. The value can be from 0 to 2 where 0 is an understanding of 0%, 1 of 50% and 2 of 100%, while a value of -1 is when the student leaves the lesson. It is tagged at the instant of the video in which the student inserts it and is intended to last until the instant of the next reaction having the same user and the same lesson. Each understanding reaction takes into account how many students are understanding 0, 1 or 2 at that particular moment, so as to be able to subsequently reconstruct the understanding of the lesson. Obviously, each understanding belongs to a user and a lesson.
- **NOTES**: represents students' notes. It is identified by a unique id and a UUID. Each note has a text and belongs to a lesson and to a student, who can share it publicly or keep it private. Furthermore, the student can associate it with a particular slide, if it exists. These notes are then collected with the possibility of downloading them in a file that creates a handout for the lesson.
- **MESSAGE**: represents messages in chat. It is identified by a unique id and a UUID. Each message has a text and is associated with the lesson and a sender. A message is categorized into a normal message (simple chat message, not tagged to the video) or question (tagged to the exact second of the video in which it is posted). If a message is a question then it also has a status, which is 0 if it is open and 1 if the sender marks the question as resolved.
- **ANSWER**: represents the answers to questions. It is identified by a unique id and a UUID. Each answer has a text and is associated with the question and a sender.
- **REACTION**: represents the reactions of LIKE (code 1), I DON'T LIKE (code 0). It is identified by a unique id and a UUID. The type of reaction is identified in the type attribute, if you want to insert new types it would be enough to use new codes in the type attribute. Each reaction can be associated with a chat message (normal message or question), an answer or a note.

Having clear the details of each entity and the necessary attributes, we can define the following logical schema:

Legend:

- **Key** : primary key
- Key : foreign key
- Attribute *: optional attribute

3.4.1. Logical mapping

USER (**email** : STRING, firstName : STRING, lastName : STRING, password: STRING, createdAt : DATE, updatedAt : DATE)

COURSE (**courseId** : NUMBER, courseUUID : UUID, name: STRING, teacherEmail : STRING, createdAt : DATE, updatedAt : DATE)

ENROLLMENT (studentEmail : STRING, courseId : NUMBER, status: NUMBER, createdAt : DATE, updatedAt : DATE)

LECTURE (**lectureId** : NUMBER, lectureUUID : STRING, status: NUMBER, youtubeVideoURL : STRING, presentationPath *: STRING, scheduledStartTime : DATE, scheduledEndTime : DATE, chatEnabled : BOOLEAN, notesEnabled : BOOLEAN, feedbackEnabled : BOOLEAN, coursed : BOOLEAN, mooc , createdAt : DATE, updatedAt : DATE)

UNDERSTANDING (**understandingId** : NUMBER, understandingUUID : UUID, value: NUMBER, videoSeconds : NUMBER, globalUp : NUMBER, globalMid : NUMBER, globalDown : NUMBER, lectureId : NUMBER, userEmail : STRING, createdAt : DATE, updatedAt : DATE)

NOTE (**noteId** : NUMBER, noteUUID : UUID, text: STRING, pageNumber *: NUMBER, private: BOOLEAN, userEmail : STRING, lectureId : NUMBER, createdAt : DATE, updatedAt : DATE)

MESSAGE (**messageId** : NUMBER, messageUUID : UUID, text: STRING, senderEmail : STRING, lectureId : NUMBER, videoSeconds *: NUMBER, status *: NUMBER, createdAt : DATE, updatedAt : DATE)

ANSWER (**answerId** : NUMBER, answerUUID : UUID, text: STRING, status: NUMBER, messageId : NUMBER, userEmail : STRING, createdAt : DATE, updatedAt : DATE)

REACTION (**reactionId** : NUMBER, reactionUUID : UUID, value: NUMBER, answerId *: NUMBER, messageId *: NUMBER, noteId *: NUMBER, userEmail : NUMBER, createdAt : DATE, updatedAt : DATE)

Note 1: Each record has two attributes createdAt and updatedAt which represent the instants in which the record is created and modified respectively.

Note 2: If the primary key is an auto-incrementing unique identifier, a UUID attribute is associated. The purpose is, for security reasons, never to reveal the primary key of the record on the client side. Also, by providing the UUID instead

of the auto-incremented ID, it is not possible to get an idea of how many records there are in the table.

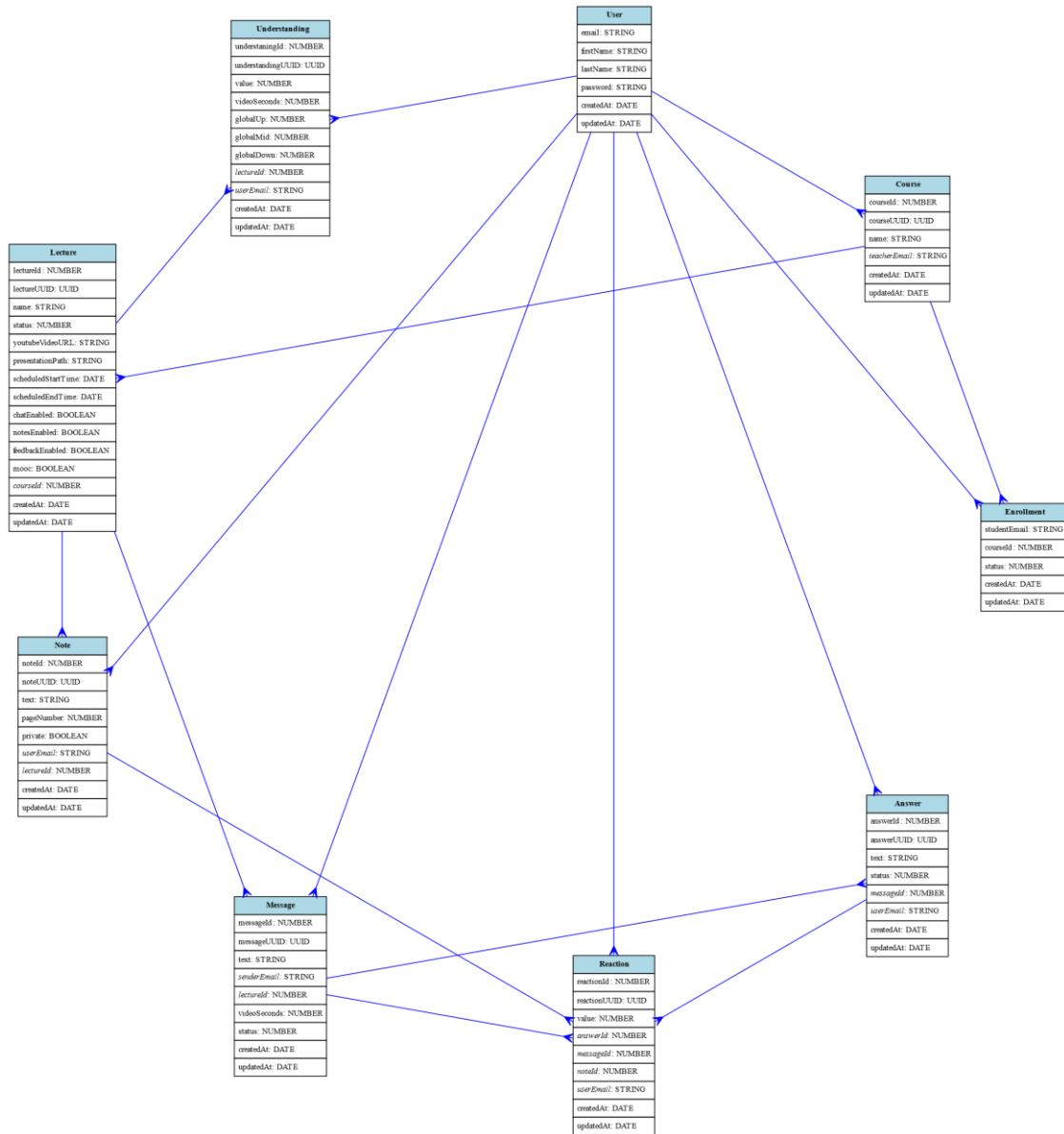


Figure 3.5: Logical schema

3.4.2. Data Access Layer

Each of the entities identified in the logical schema is mapped by Sequelize to a model class. Each instance of the model class represents an object of that model that maps a row of the table to the database. In this way, the model instances are Data Access Objects (DAOs). From this instance, it is in fact possible to query the database.

3.5. Application Tier

3.5.1. Application structure

First of all, we will see how the source folder of the server is structured, then we will analyze each part.

```

src
|  app.js           # App entry point
|  __api            # Express routes controllers for all the endpoints of the app
|  __client        # Main folder of React client application
|  __config        # Configuration related files
|  __loaders       # Split the startup processes into modules
|  __models        # Database sequelize models
|  __services      # All the business logic is here
|  __socketServer  # Socket server
|  __.env          # Environment variables

```

Figure 3.6: Server side application structure

3.5.2. Services

Service classes encapsulate the business logic of the application and separate it from the rest of the application. Each class contains functions useful for processing and extracting data, collected through the application's data access layer. Service classes interface controllers to data access.

Let's now see the service classes with their functions:

USER SERVICE

Table 3.1: User service functions

| Name | Description |
|-----------------------|--|
| getUserByEmail | Given an email, it returns the user with such email. |
| createNewUser | Create a new user, receiving in input the email, first name, last name and password. Then it hashes the password and return the user just created. |

COURSE SERVICE

Table 3.2: Course service functions

| Name | Description |
|------|-------------|
|------|-------------|

| | |
|---------------------------|---|
| createCourse | Create a new course given its name and the email of the teacher. Then it returns the course just created. |
| getCourseByUUID | Given an UUID, it returns the course with such UUID. |
| getTaughtCourses | Given the teacher email, it returns the list of courses taught by the teacher. |
| getFollowedCourses | Given the student email, it returns the list of courses followed by the student. |
| deleteCourseByUUID | Delete a course given its UUID. |
| enrollStudent | Enroll a student to a course. |
| deleteEnrollment | Delete the enrollment of a student to a course. |

LECTURE SERVICE

Table 3.3: Lecture service functions

| Name | Description |
|------------------------------|--|
| createLecture | Create a new lecture given its name, scheduled start time, scheduled end time, membership course and optionally the video URL and presentation path. |
| getLectureByUUID | Given an UUID, it returns the lecture with such UUID. |
| getLectureByID | Given an id, it returns the lecture with such id. |
| getLecturesByCourseID | Given the id of a course, it returns the lectures of such course. |
| deleteLectureByUUID | Delete the lecture with the given UUID. |
| getYoutubeLiveVideo | Get the list of YouTube live broadcasts of the Google User using the YouTube Data API. |

| | |
|-------------------------------------|---|
| getYoutubeChannels | Get the list of YouTube channels of the Google user using the YouTube Data API |
| getLecturesAvailableForMoocs | Get the list of the lecture that are open, ie the teacher has marked them as available for MOOCs. |

CHAT SERVICE

Table 3.4: Chat service functions

| Name | Description |
|--------------------------------|--|
| saveMessage | Create and save a new message given its text, the sender and the referring lecture id. |
| saveQuestion | Create and save a new question message given its text, the sender, the video seconds in which it is posted and the sender. |
| solveQuestion | Change the question status from OPEN (0) to SOLVED (1). |
| getMessageByUUID | Given an UUID, it returns the message with such UUID. |
| getMessagesByLectureID | Given a lecture id, it returns the messages of the chat of such lecture. |
| getQuestionsByLectureID | Given a lecture id, it returns only the question messages of the chat of such lecture. |
| getMessageById | Given an id, it returns the message with such id. |

FEEDBACK SERVICE

Table 3.5: Feedback service functions

| Name | Description |
|------|-------------|
|------|-------------|

| | |
|--------------------------------|--|
| getAnswerByUUID | Given an UUID, it returns the answer with such UUID. |
| getNoteByUUID | Given an UUID, it returns the note with such UUID. |
| getNoteById | Given an id, it returns the note with such id. |
| saveAnswer | Create and save a new answer given its text, the sender and the referring message id. |
| saveNote | Create and save a new note given its text, the tagged page number if it exists, the visibility (private or shared), the user email and the referring lecture id. |
| getNotesByLectureId | Given a lecture id, it return all the notes of such lecture. |
| getUserNotesByLectureId | Given a lecture id and a user email, it return all the notes written by the user and referring of that lecture. |
| saveAnswerReaction | Add and save a new reaction to an answer given the answer id, the user email and the reaction value (0 - DISLIKE, 1 - LIKE). |
| saveMessageReaction | Add and save a new reaction to a message given the message id, the user email and the reaction value (0 - DISLIKE, 1 - LIKE). |
| saveNoteReaction | Add and save a new reaction to a note given the note id, the user email and the reaction value (0 - DISLIKE, 1 - LIKE). |
| getAnswerReaction | Given a user email and an answer id, it returns the reaction of such user to the answer. |
| getNoteReaction | Given a user email and a note id, it returns the reaction of such user to the note. |

| | |
|---|--|
| getMessageReaction | Given a user email and a message id, it returns the reaction of such user to the message. |
| saveUnderstanding | Create and save a new understanding reaction given its value (-1, 0, 1, 2), the count of person currently understanding 0, 1 and 2, the lecture id and the user email. |
| getLectureUnderstandings | Given a lecture id it returns all the understanding reactions of that lecture. |
| getStudentLecture Understandings | Given a lecture id and a user email, it return all the understanding reactions of that user in that lecture. |

3.5.3. API

In this section we will analyze the Express routes, indicating the path in which the different endpoints are located and the description of what each API returns. Furthermore, we will understand the process of authentication through JWT and how, in the protected routes, the application guarantees that the user is logged.

In the application there are 3 main Express routes:

Table 3.6: Express routes

| Route | Description |
|-------------------|----------------------------|
| / auth | Authentication related API |
| / courses | Courses related API |
| / lectures | Lecture related API |

For each route, we will analyze its endpoints, describing the path in which they are located, the http method, the description and the JSON returned in case of success or in case of fail.

AUTH ROUTE

Table 3.6: Auth route endpoints

| Path | Method | Description | Success | Fail |
|------|--------|-------------|---------|------|
|------|--------|-------------|---------|------|

| | | | | |
|---------------------|------|--|-------------------------------------|---|
| /register | POST | Register a new user. | { registerStatus : 'ok'} | { registerStatus : 'ko', error: errors} |
| /login | POST | Log in to user. If the login succeed returns the user. Save the access token in the cookies. | { loginStatus : 'ok', user: user} | { loginStatus : 'ko', message: 'Wrong credentials'} |
| /logout | GET | Logout the user | { logoutStatus : 'ok'} | / |
| /editProfile | POST | Edit the profile info | { profileStatus : 'ok', user: user} | { profileStatus : 'ko', errors: errors} |

COURSES ROUTE

Table 3.7: Course routes endpoints

| Path | Method | Description | Success | Fail |
|--------------------|--------|---|------------------|--------|
| /instructor | GET | Returns the list of courses taught by the logged user | Taught courses | / |
| /followed | GET | Returns the list of courses followed by the logged user | Followed courses | / |
| /newCourse | POST | Create a new course taught by the logged user | Course created | Errors |

| | | | | |
|----------------------------------|--------|---|--------------------|--------|
| <u>/: uuid</u> | DELETE | Delete a course given its uuid | Course deleted | Errors |
| <u>/: uuid</u> | GET | Return a course given its uuid | Course | Errors |
| <u>/ enroll</u> | POST | Enroll a student to a course | Course | Errors |
| <u>/enroll/:courseUID</u> | DELETE | Delete a student enrollment to a course | Deleted enrollment | Errors |

LECTURES ROUTE

Table 3.8: Lectures route endpoints

| Path | Method | Description | Success | Fail |
|---|--------|--|-----------------|--------|
| / newLecture | POST | Create a new lecture | Created lecture | Errors |
| / updateLecture | POST | Update a lecture | Updated lecture | Errors |
| / removePresentation | POST | Remove the presentation from a lecture | Updated lecture | Errors |
| / courseLectures /: courseUUID | GET | Get the lectures list of a course | Lectures | Errors |
| / getLecture /: lectureUUID | GET | Get a lecture with a given UUID | Lecture | Errors |
| / chat /: lectureUUID | GET | Get the chat messages of a lecture with a given UUID | Messages | Errors |

| | | | | |
|---|-----|--|-----------------|--------|
| / understandings /: lectureUUID | GET | Get the understanding reactions of a lecture with a given UUID | Understandings | Errors |
| / studentUnderstandings /: lectureUUID | GET | Get the understanding reaction of the logged user in a lecture with a given UUID | Understandings | Errors |
| / notes /: lectureUUID | GET | Get the notes of a lecture with a given UUID | Notes | Errors |
| / deleteLecture /: uuid | GET | Delete the lecture with the given uuid | Deleted lecture | Errors |
| / downloadPDF /: uuid | GET | Start the download of the presentation of the lecture with the given uuid | File | Errors |
| / downloadNotes /: uuid | GET | Start the download of the notes of a lecture with the given uuid | File | Errors |
| / channels | GET | Retrieve the list of YouTube channels of the logged user with Google | Channels | null |

| | | | | |
|-------------------------|-----|---|-------|------|
| / liveBroadcasts | GET | Retrieve the lists of YouTube live broadcast of the logged user with Google | Video | null |
|-------------------------|-----|---|-------|------|

3.5.4. Authentication

JWT¹², acronym for JSON Web Token, is an encryption and contact system in JSON format for exchanging information between the various services of a server. This generates a token that can be encrypted and signed using a key available only to the person who actually generated it.

The signature algorithm is processed through HMAC or with public and / or private keys with RSA or ECDSA standard.

JWTs are now a consolidated reality and are used in web services to authenticate a client. The operating system is quite simple: the client sends a request to the server and this generates an authentication token that the client will use every time it connects to the same node.

The token is made up of three essential bodies:

Header that contains the two basic information: the type of token, which in our case is JWT, and that of the algorithm used for encryption;

Payload, the block that contains the exchange information between the parties;

Signature, the encryption phase of the two parts: Payload and Header . These are first merged and then subjected to an extremely complex encryption operation. At the end of the operation a key is generated which will give rise to a token of over 200 characters.

However, it is important to understand that the token content is not encrypted. It can be unpacked and decrypted at will so it is important that no sensitive data such as passwords are entered. Security is guaranteed with a secret key from the server and therefore any data that is manipulated will not be accepted by the latter, effectively prohibiting access to the data.

In EVOLI Sync the token is created at login. The token payload will contain information about the user: email, name and surname. Once encrypted with a secret key contained in the server, the token is added to cookies with a maximum duration

¹² From <https://psicografici.com/jwt-json-web-token/>

of one day. Later it will be exchanged in every communication between client and server. Upon logging out, the cookie will come deleted .

3.5.5. Socket communication

A large part of the communication between client and server is made by socket and the socket.io js library. In particular, all the user's interaction during the live lecture and report pages is completely implemented with a socket communication.

Socket.IO¹³ is a library that enables real-time, bidirectional and event-based communication between the browser and the server. It consists of:

- A Node.js server
- A Javascript client library for the browser (which can be also run from Node.js)

The bidirectional channel between the Socket.IO server (Node.js) and the Socket.IO client (browser, Node.js, or another programming language) is established with a WebSocket connection whenever possible, and will use HTTP long-polling as fallback.

Our socket architecture is room-based. For each lecture we create a room identified by the lecture UUID. Each room contains the information about the users currently connected to the lecture and the instructor. For each user, the current understanding value is saved. In the moment the student joins a lecture, its default understanding value is 2 and when he leaves the lecture the understanding value is -1. In this way, we are able to reconstruct the history of understanding of a user during in the lecture.

Each time the user enter in the report page or in the lecture page (both instructor page and student page) a socket connection is established with the server, then, when the user leave the back, the socket channel is disconnected.

Now, we will list and describe all the events exchanged between client and server.

Table 3.9: Socket events

| Event | Direction | Description | Payload |
|----------------|-----------|---|---------|
| connect | C->S | A user establishes a socket connection with the server. | / |

¹³ From <https://socket.io/docs/v4/>

| | | | |
|--------------------------------|--------|--|---|
| joinLecture | C -> S | A student joins the lecture, he is added to the room. Then he will receive all the lecture resume data: chat, notes, participants , functionalities status. | user, room |
| instructorJoin Lecture | C-> S | An instructor joins the lecture, he is added to the room. Then he will receive all the lecture resume data: chat, notes, participants, functionalities status. | user, room |
| report | C-> S | A user wants to see the report of a lecture. He will receive the lecture info such as questions, notes and functionalities status. | user, room |
| participants | S-> C | The list of participants is sent to the client. | Participants, instructor |
| chatResume | S-> C | Send all the chat messages of the lecture. | Messages |
| questionsResume | S-> C | Send all the questions posted in the chat of the lecture. | Questions |
| notesResume | S-> C | Send all the notes posted during the lecture. | Notes |
| newFunctionality Status | S-> C | Send the functionalities status, specifying whether the chat is enabled, the notes are enabled and the feedbacks are visible to the students. | chatActive , notesActive , understanding Active |
| understanding | C-> S | A student change his understanding value. It is saved into the db through the feedback service, then the understandings are sent back as update. | room, understanding, videoSeconds , user |

| | | | |
|-----------------------------|-------|--|---------------------------------------|
| Understanding Change | S-> C | The understanding status of the lecture is sent to the client. It will keep track how many students are understanding 0, 1 or 2. | [# 0, # 1, # 2] |
| question | C-> S | A user posts a question in chat. It is saved in the db through the chat service and then the question is sent to the whole room. | Question, user, room, videoTime |
| newChatMessage | S-> C | Send the message to the client, it can be a normal message or a question. | |
| answer | C-> S | A user posts an answer to a question. It is saved into the db through the feedback service, then the chat messages are sent. | answerText , messageUUID , user, room |
| chatMessage | C-> S | A user posts a normal chat message. It is saved in the db through the chat service and then the message is sent to the whole room. | Message, sender, room |
| solveQuestion | C-> S | A user marks an own question as solved. The question status is updated through the chat service the update is broadcasted to the whole room. | messageUUID , room |
| messageUpdate | S-> C | Sends an updated chat message to the client. | Message |
| sharedNote | C-> S | A user publish a new shared note. It is saved in the db through the feedback service, then it is broadcasted to the whole room. | Notes, slideNumber , user, room |
| privateNote | C-> S | A user publish a new private note. It is saved into the db through the | Notes, slideNumber , user, room |

| | | | |
|------------------------------|-------|---|---|
| | | feedback service, then it is broadcasted to the whole room. | |
| newNote | S-> C | A note is sent to a client. | Note |
| reaction | C-> S | A user insert a reaction on either an answer, a message or a note. The reaction is saved (or updated) into the db and then the object on which it is made is broadcasted to the whole room. | Notes, slideNumber , user, room |
| noteUpdate | S-> C | Sends an updated note to the client. | Note |
| changePage | C-> S | The instructor change the page number of the presentation. The page number is then broadcasted to the students. | pageNumber |
| instructorChange Page | S-> C | The server send the current instructor page number to the client. | pageNumber |
| Functionality Status | C-> S | The instructor activates / disactivates a functionality (chat, notes, feedback visibility). The lecture is updated and the information is sent back to the whole room members. | chatActive , notesActive , understanding Active. Room |
| endLecture | C-> S | The instructor ends the lecture. The scheduled end time is updated. | Room |
| InstructorEnd Lecture | S-> C | The server notifies the room members that the instructor has ended the lecture. | / |
| leaveLecture | C-> S | A student leaves the lecture. The participants list and the understandings are sent to the whole room. If the room is empty it is eliminated. | Room |

| | | | |
|--------------------------------|------|--|------|
| instructorLeave Lecture | C->S | An instructor leaves the lecture. The participants list is sent to the whole room. If the room is empty it is elinated . | room |
| disconnect | C->S | A client disconnect the stablished socket connection. | / |

3.6. Presentation tier

3.6.1. React application

First of all, we will see how our client application folder is composed:

```
src
|__assets      # Fonts, images folder
|__components  # React components divided by functionality
|__context     # React context API, create a reusable store
|__i18n        # Language translations
|__pages       # Client application pages
|__App.js      # React app file
```

Figure 3.7: React application structure

3.6.2. Routes and pages

React Router¹⁴ is a fully-featured client and server-side routing library for React, a JavaScript library for building user interfaces. React Router runs anywhere React runs; on the web, on the server with node.js, and on React Native.

EVOLI Sync uses React Router to give the user the perspective to have multiple pages in a single page React app.

Each one of our pages is bound to a route, in particular we have:

Table 3.10: Mapping between React routes and pages

| Route | Protected | Page component |
|-------|-----------|----------------|
| / | No | IndexPage |

¹⁴ From <https://github.com/remix-run/react-router/blob/main/docs/getting-started/tutorial.md>

| | | |
|--------------------|-----|-----------------------|
| / about | No | AboutPage |
| / home | Yes | Home page |
| / mooc | No | MoocPage |
| / studentLecture | Yes | StudentLecturePage |
| / lecture | Yes | InstructorLecturePage |
| / report | Yes | ReportPage |
| / instructorReport | Yes | ReportInstructorPage |
| / error | No | ErrorPage |
| Other | No | NotFound |

PROTECTED ROUTES

Protected routes are simply routes that are not accessible when the user is not logged. In particular, if the user is not logged then he is redirected on the / route, where the login form is present.

3.6.3. Site Map

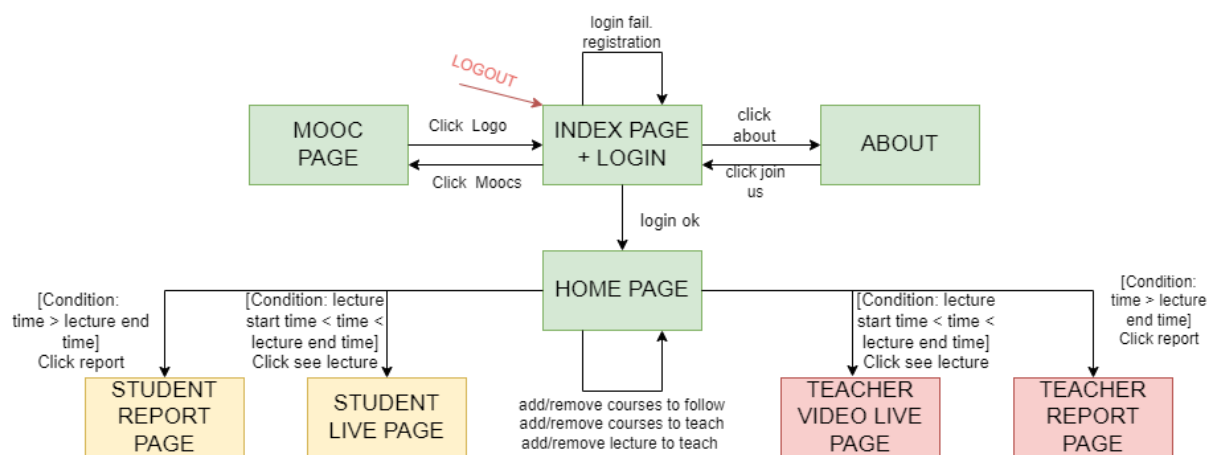


Figure 3.8: Page graph

TEACHER WORKFLOW EXAMPLE

In this paragraph we are going to analyze the basic workflow of how a teacher creates a lesson and follows its progress live. Once finished, it goes to view the report.

1. The user, not yet logged in, is on the INDEX PAGE which also contains the login form
2. The user enters his email credentials and password and is redirected to the HOME PAGE, where he enters the teacher's section.
3. The HOME PAGE shows the list of courses taught by the teacher and clicking on a course displays the list of lessons. At this point, press the button to create a new lesson and then a modal appears where you can enter the lesson data.
4. The teacher inserts the name of the lesson, the date and time of departure, the date and time of the end, the URL of the live YouTube and possibly the PDF slides of the lesson. He also chooses whether to make the lesson available as a mooc.
5. Once the teacher enters all the necessary information, the button appears that allows the teacher to go to the LIVE lesson page.
6. The teacher is on his lecture page where he can see the lecture video, feedback dashboard, PDF presentation, chat and notes.
7. The teacher, when he wants to end the lesson, clicks on the button to end the lesson and is redirected to the HOME PAGE, where the button to view the report of the lesson is now visible.
8. The teacher clicks on the REPORT button and is redirected to the report page.

STUDENT WORKFLOW EXAMPLE

In this paragraph we are going to analyze the workflow of a student who accesses the platform, follows a live lesson and at the end sees the report.

1. The user, not yet logged in, is on the INDEX PAGE which also contains the login form .
2. The user enters his email credentials and password and is redirected to the HOME PAGE, where he enters the student section.
3. The HOME PAGE shows the list of courses followed by the student and clicking on a course displays the list of lessons. At this point he selects a lesson, which is currently live, and presses the button that allows the student to go to the lesson page.
4. The student follows the lesson live by viewing the lesson video, PDF presentation, feedback dashboard, chat and notes.
5. When the teacher finishes the lesson, the student is redirected to the HOME PAGE where the button to see the lesson report is now available,

- The student clicks on the REPORT button and is redirected to the report page.

3.6.4. Mockups

This section shows the mockups of the pages of the platform with a brief description of what is on the page.

INDEX PAGE

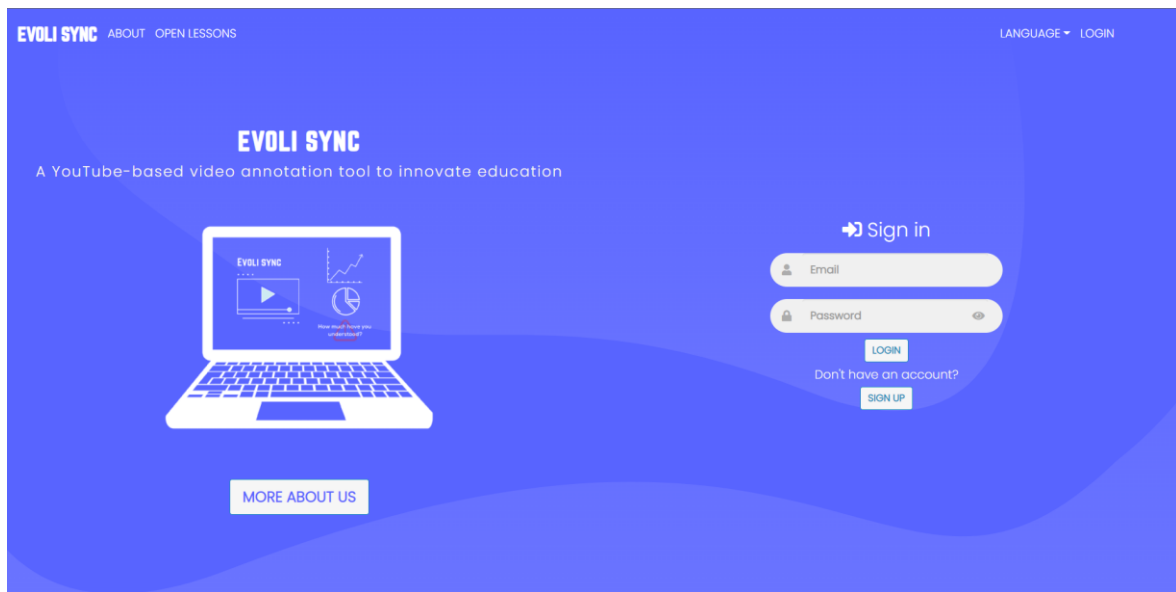


Figure 3.9: Index page

The INDEX page shows a quick text description of the platform with the login / registration form . The user logs in by entering an email and password and pressing the LOGIN button. To register, the user clicks on the SIGN UP button, at this point the registration form appears in which the user enters email, name, surname, password and confirmation password. Complete the registration by accepting the privacy policy and clicking on the confirm button. Furthermore, the page contains a MORE ABOUT US button to redirect the user to the ABOUT page.

Figure 3.10: Register form

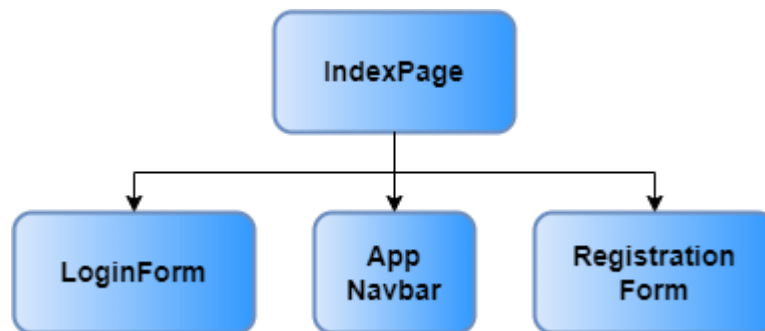


Figure 3.11: Index page components

ABOUT PAGE

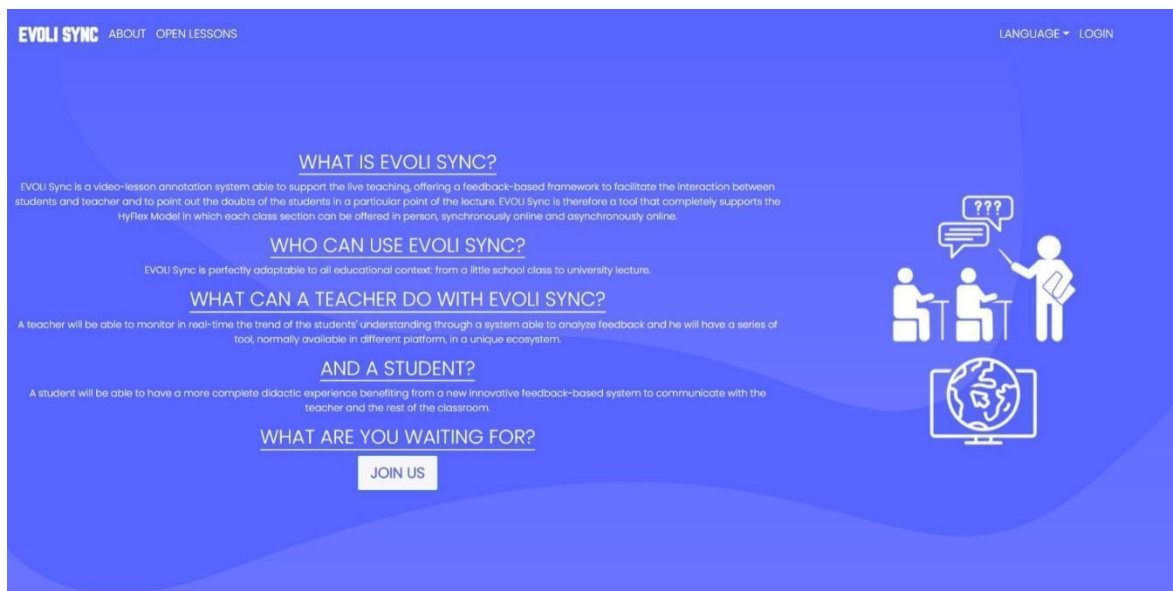
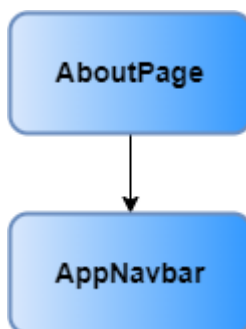


Figure 3.12: About page



The ABOUT page shows a series of EVOLI Sync information: a brief description of the platform, which users it is addressed to, what a teacher and student can do with EVOLI Sync.

Figure 3.13: About page components

HOME PAGE

The home page is divided into two sections: the teacher section and the student section.

The teacher section shows the list of courses taught with the name and creation date. For each course, it is possible to copy the code, delete it and see the participants using the appropriate buttons. Furthermore, the teacher can add a new course by pressing the CREATE A NEW COURSE button which brings up a form in which to insert the name of the course to be created.

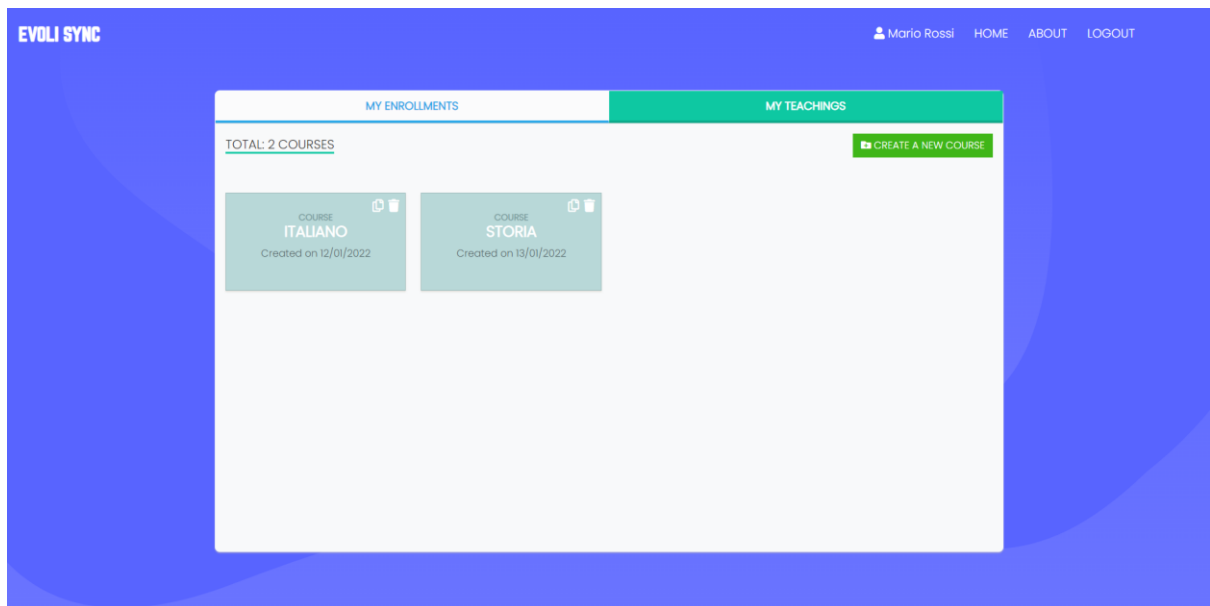


Figure 3.14: Home page teacher section

Once a course has been selected, a table with the list of lessons appears. For each lesson, the name, date, time, status and a series of buttons are displayed that allow you to: modify the lesson if it is not finished, delete the lesson, download the slides (if they exist), download the notes and view the report (if the lesson is finished).

Create a new lecture

Name*
Enter name

Scheduled start time*
08/03/2022 11:48

Scheduled end time*
gg/mm/aaaa --:--

[START A YOUTUBE LIVE](#)

YouTube video URL
[Empty field]

[TEST VIDEO](#)

PDF Presentation
Scegli il file Nessun file scelto

Open lesson

[SUBMIT](#) [CLOSE](#)

Update lecture

Name*
Promessi sposi

Scheduled start time*
08/03/2022 11:53

Scheduled end time*
08/03/2022 16:00

[START A YOUTUBE LIVE](#)

YouTube video URL
https://youtu.be/r2HQcD-KN7A

[TEST VIDEO](#)

PDF Presentation
Scegli il file Nessun file scelto

Open lesson

[SUBMIT](#) [CLOSE](#)

Figure 3.15: Lecture modals

It is also possible to create a new lesson by clicking on the appropriate CREATE A NEW LECTURE button which displays a modal that allows you to enter the information of the new lesson to be created.

EVOLI SYNC HOME ABOUT OPEN LESSONS LANGUAGE Mario Rossi LOGOUT

MY ENROLLMENTS

Informatica
[Mario Rossi] - Created on 02/22/2022

MY TEACHINGS

[CREATE LECTURE](#)

[BACK TO COURSES](#)

| Lecture | Day | Timetable | Status | Actions |
|----------------|---------------|---------------------|--------|--|
| Linguaggio C | Feb 23rd 2022 | 4:00 pm - 8:00 pm | ENDED | REVIEW EDIT DELETE |
| Linguaggio C++ | Feb 23rd 2022 | 6:08 pm - 9:07 pm | ENDED | REVIEW EDIT DELETE |
| Java | Feb 24th 2022 | 10:43 am - 12:44 pm | ENDED | REVIEW EDIT DELETE |
| Socket | Feb 24th 2022 | 1:04 pm - 6:04 pm | ENDED | REVIEW EDIT DELETE |

Figure 3.16: Home page, teacher's lesson list

Similarly, the student section shows the list of courses taken with name, teacher and creation date. For each course it is possible to unsubscribe. It is also possible to subscribe to a new course by creating on the ADD A COURSE button, which brings up a form to enter the code of the course to subscribe to.

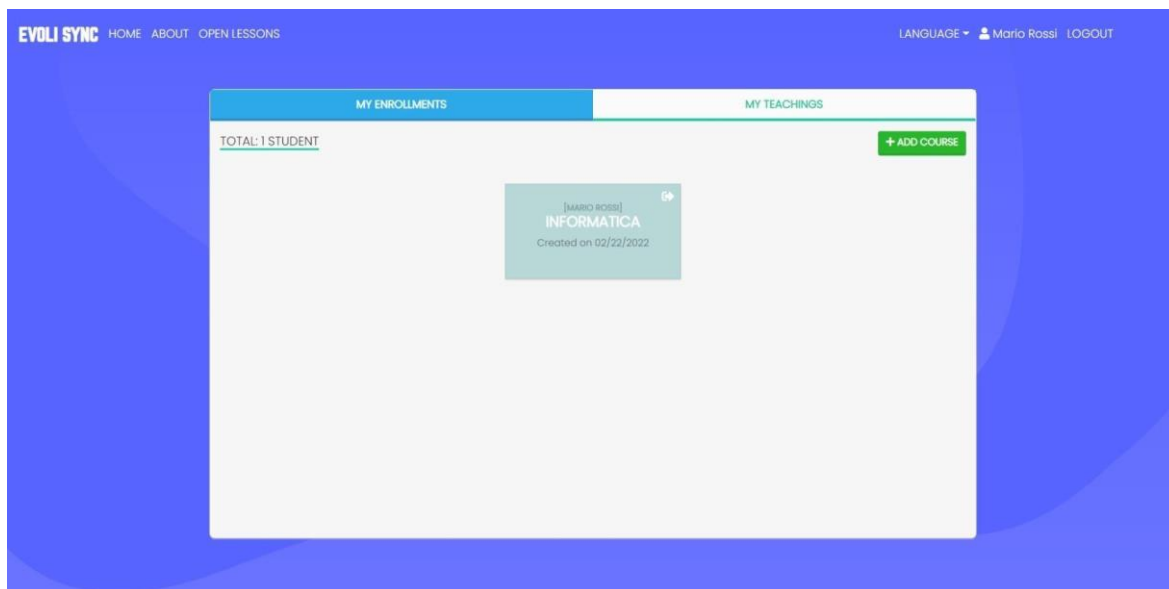


Figure 3.17: Home page, student section

Once a course has been selected, a table with the list of lessons appears. For each lesson, the name, date, time, status and a series of buttons are displayed that allow you to: download the slides (if they exist), download the notes and view the report (if the lesson is finished).

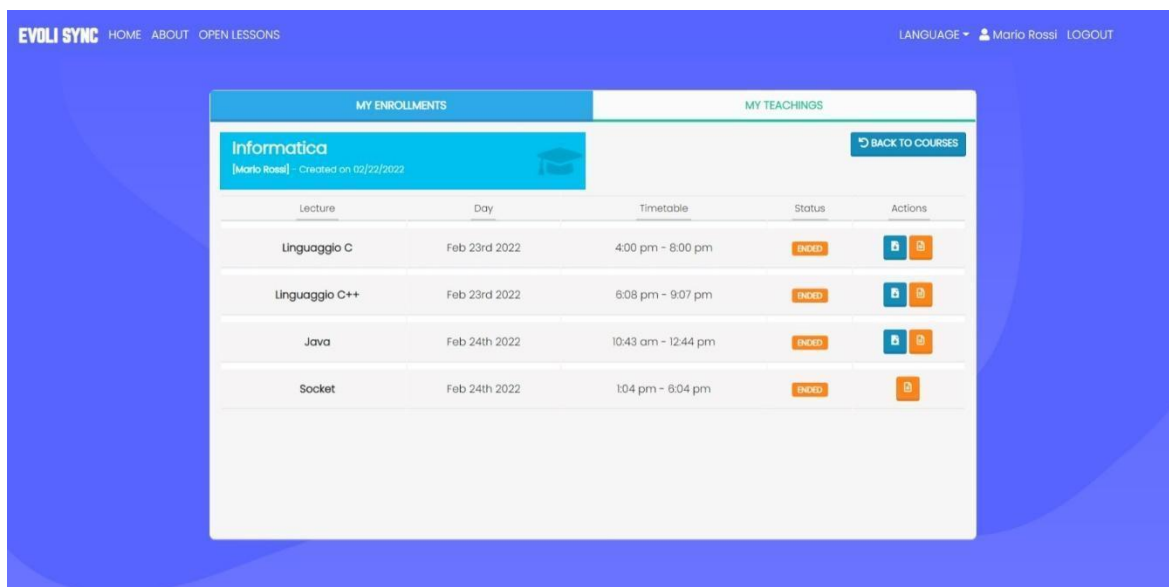


Figure 3.18: Home page, student's lesson list

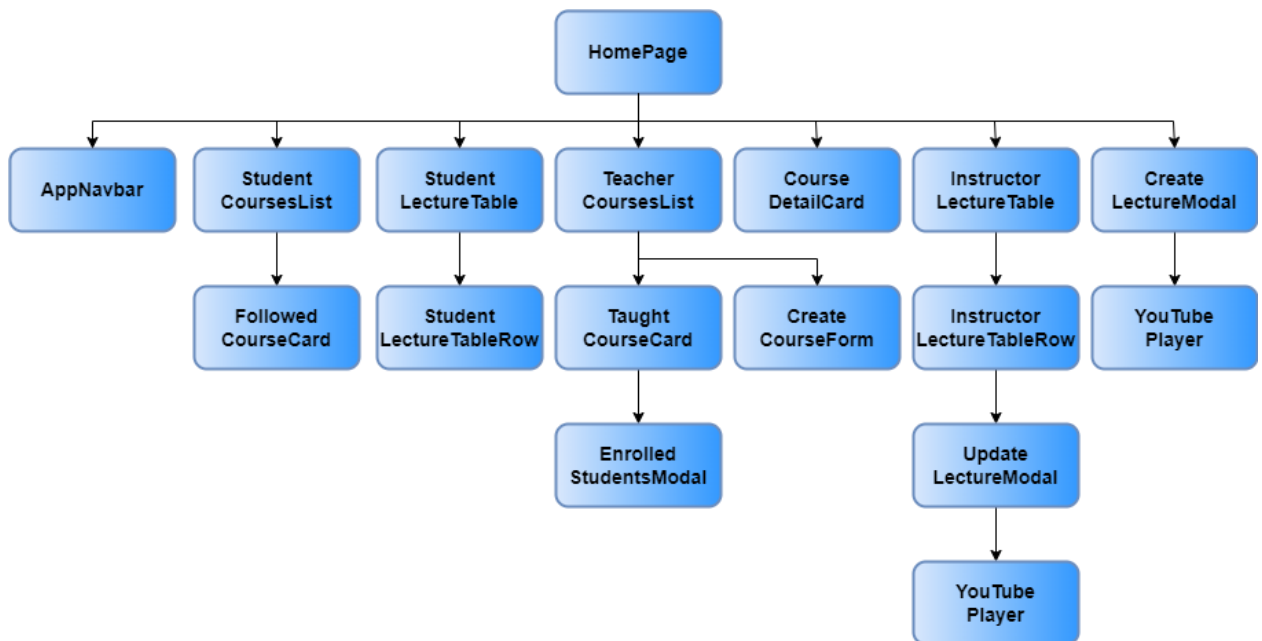


Figure 3.19: Home page components

LIVE LECTURE PAGE - STUDENT

We now show a mockup for student viewing of the lesson. Let's describe the various components that make it up:

- The video player is the YouTube player that shows the live broadcast of the lesson.
- The UNDERSTANDING panel contains a switch with 3 levels from 0 to 2 with which the student can express his degree of understanding related to the current moment of the video.
- The slide presentation is contained in a panel with the commands to change the page, zoom, and return to the slide in which the teacher is.
- The FEEDBACK panel shows a pie chart showing the number of students who are understanding 0, 1 and 2, while the line graph shows the percentage of students understanding in the last 5 minutes .
- The communication panel contains:
 - o The CHAT panel that shows both normal messages and questions tagged to the video.
 - o The NOTES panel that shows the list of notes tagged to the current slide if the slides exist or the complete list of notes if the slides do not exist. It is possible to create a new note using the appropriate blue button, by pressing on this button a text editor is displayed that allows the insertion of the new note. The user can choose whether to keep the note private by clicking on the KEEP PRIVATE button or on the SHARE button making it public.

- The PARTICIPANTS panel shows the list of participants of the lesson, with their names and surnames.

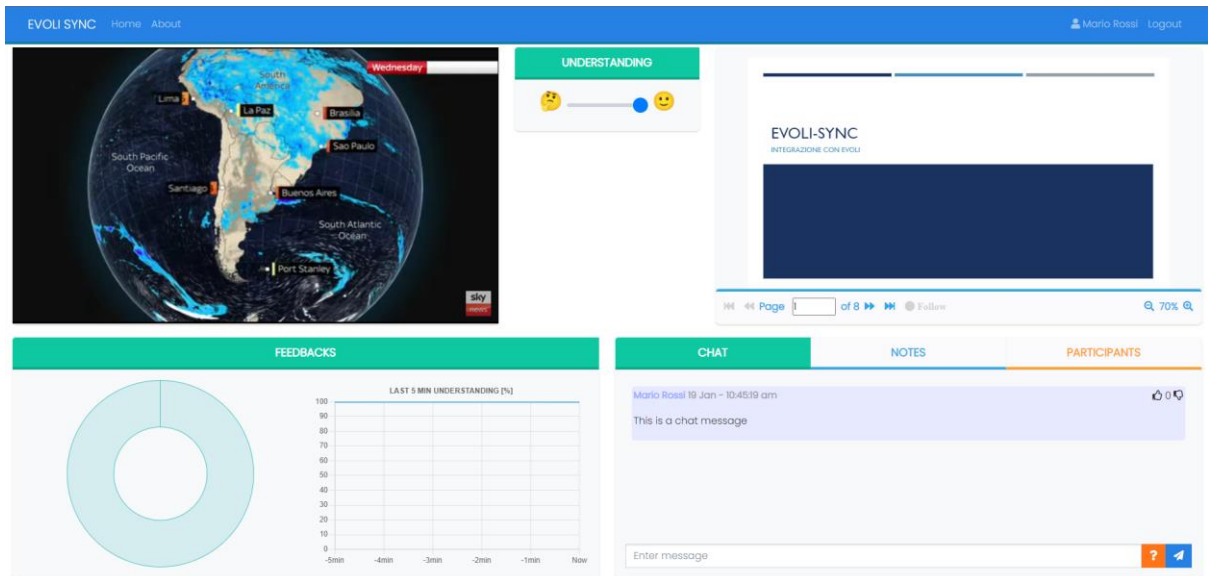


Figure 3.20: Student lecture page

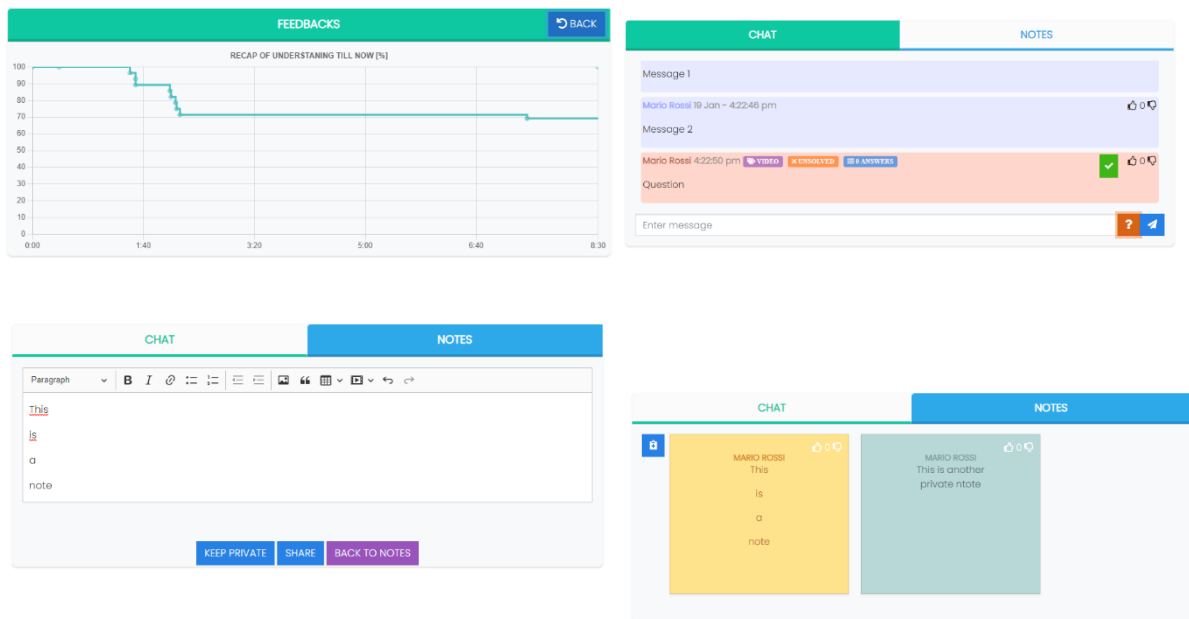




Figure 3.21: Lecture panels

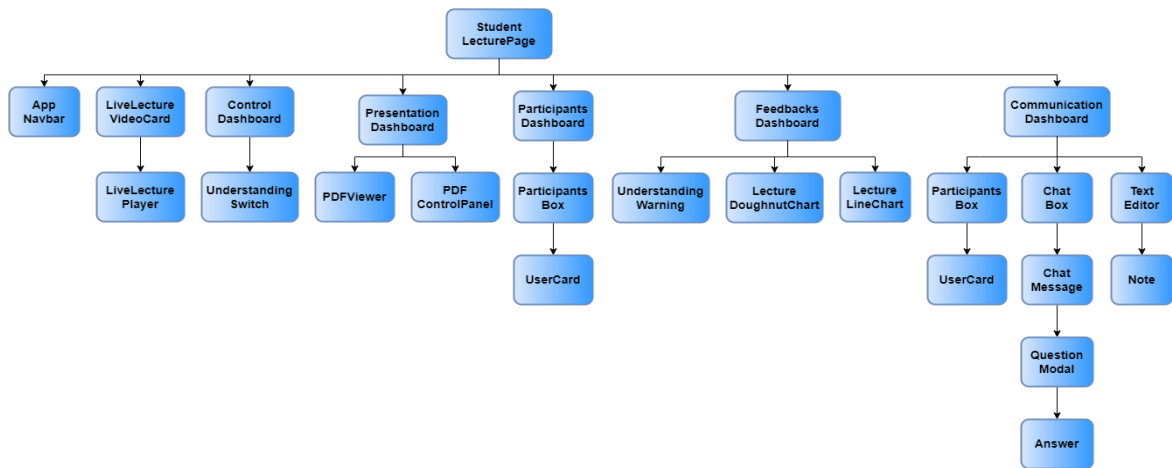


Figure 3.22: Student lecture page components

LIVE LECTURE PAGE - TEACHER

The teacher direct page has some important differences from that of the student. Obviously the UNDERSTANDING panel is not visible but a control panel is visible that allows the teacher to enable / disable chat, notes and the display of feedback to students. The FEEDBACK panel shows a RECAP button with which it is possible to see a line graph showing the trend of the student understanding percentage from the beginning of the live broadcast up to the current viewing time.

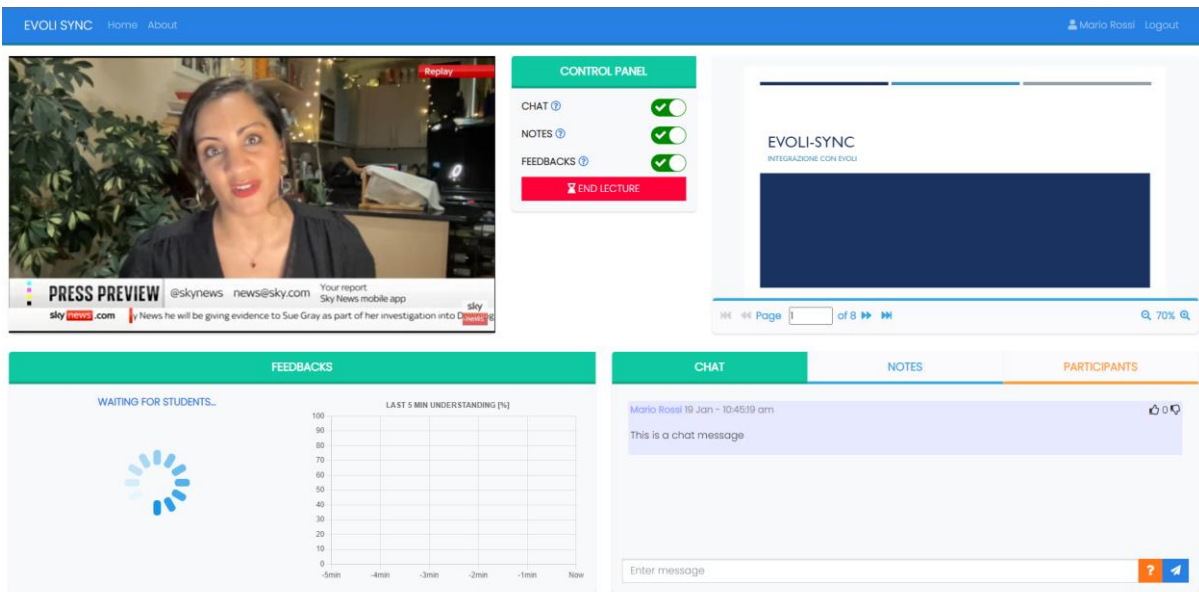


Figure 3.23: Teacher lecture page

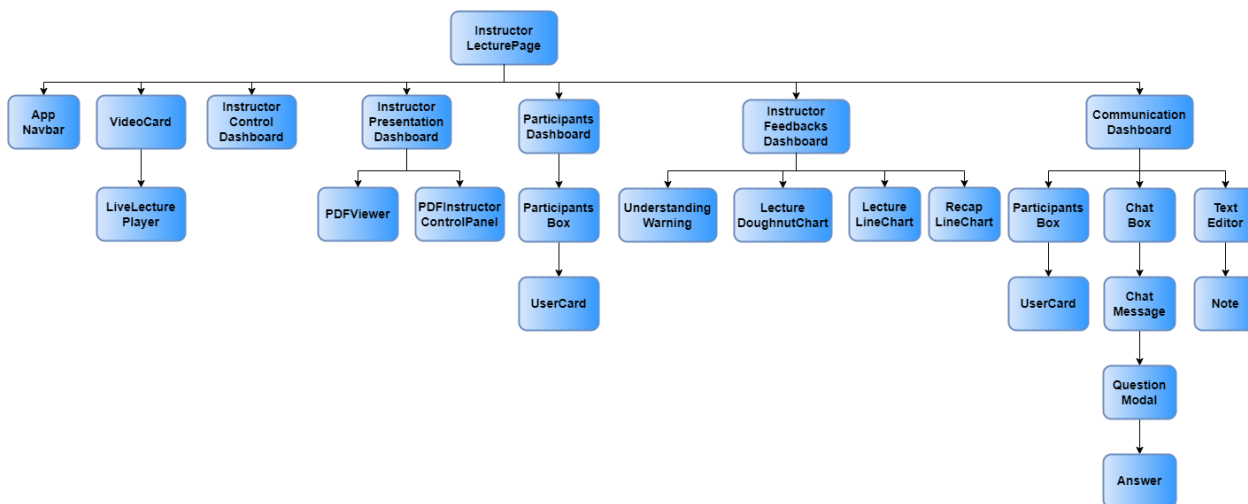


Figure 3.24: Teacher lecture page components

STUDENT REPORT PAGE

The student report page displays the video, PDF presentation, feedback report dashboard, questions, and notes.

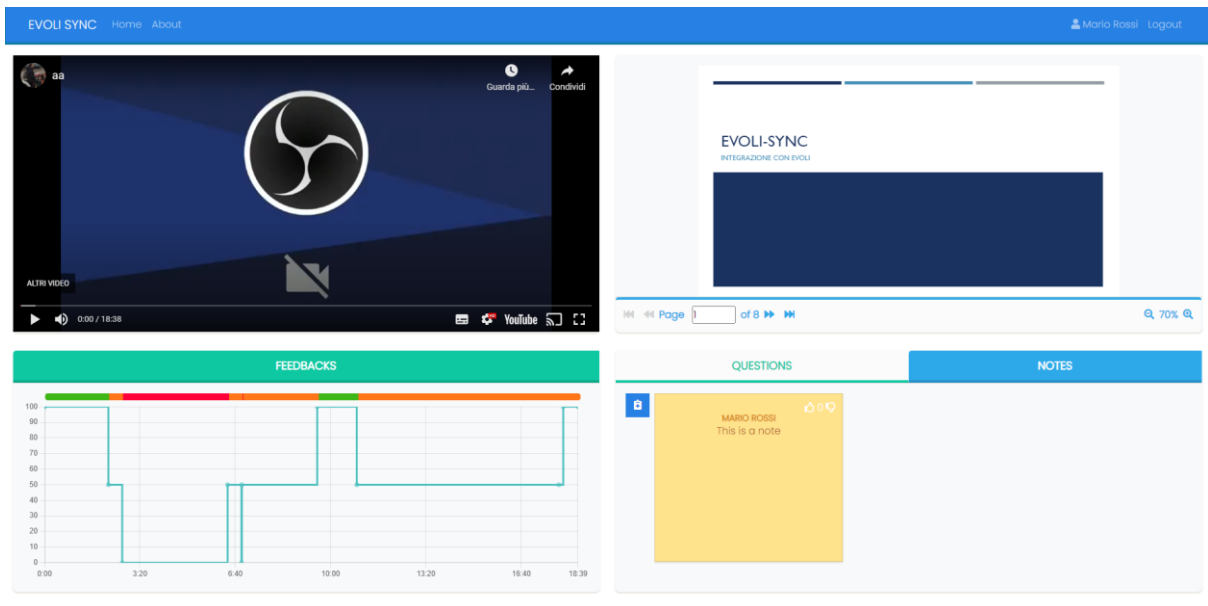


Figure 3.25: Students report page

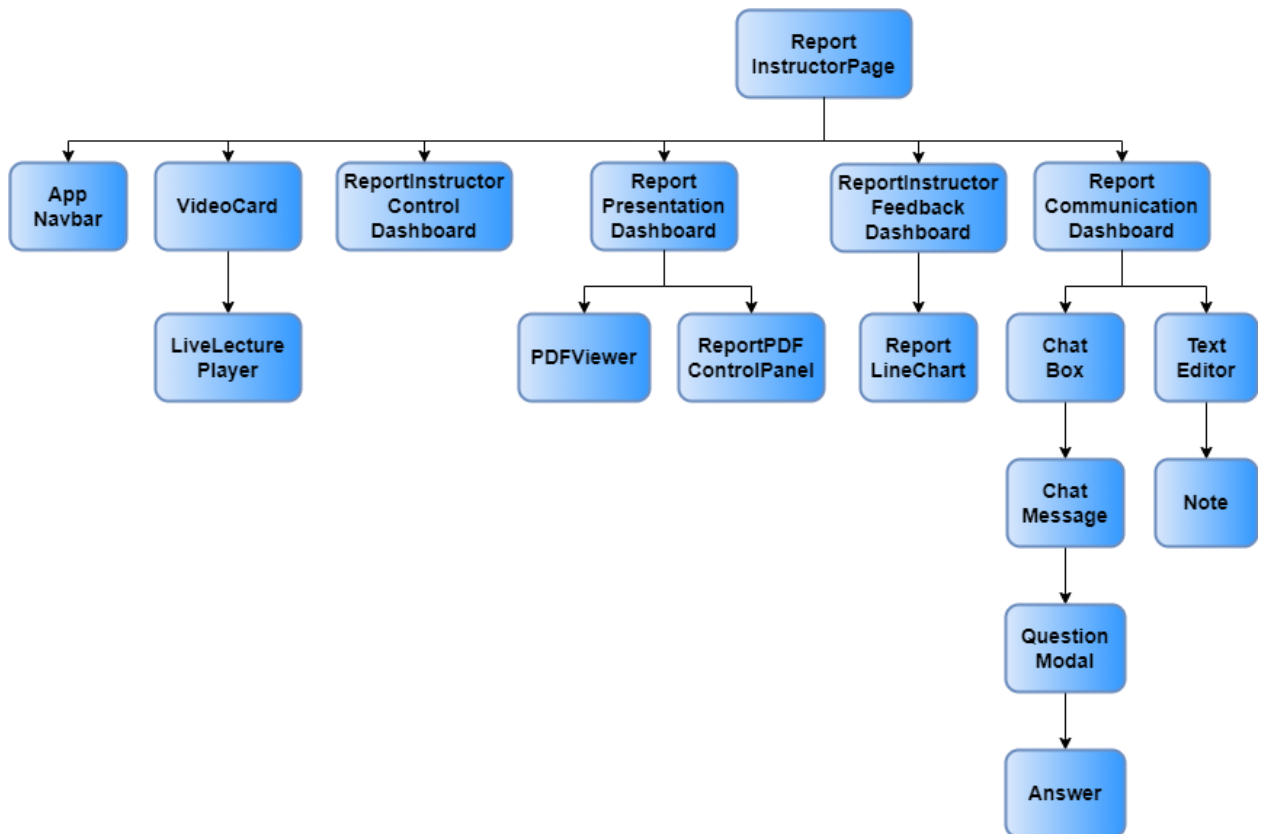


Figure 3.26: Students report page components

REPORT PAGE - TEACHER

The teacher report page shows the same functionality as the student one and, in addition, a control panel where the teacher can disable the display of questions, notes and feedback summary.

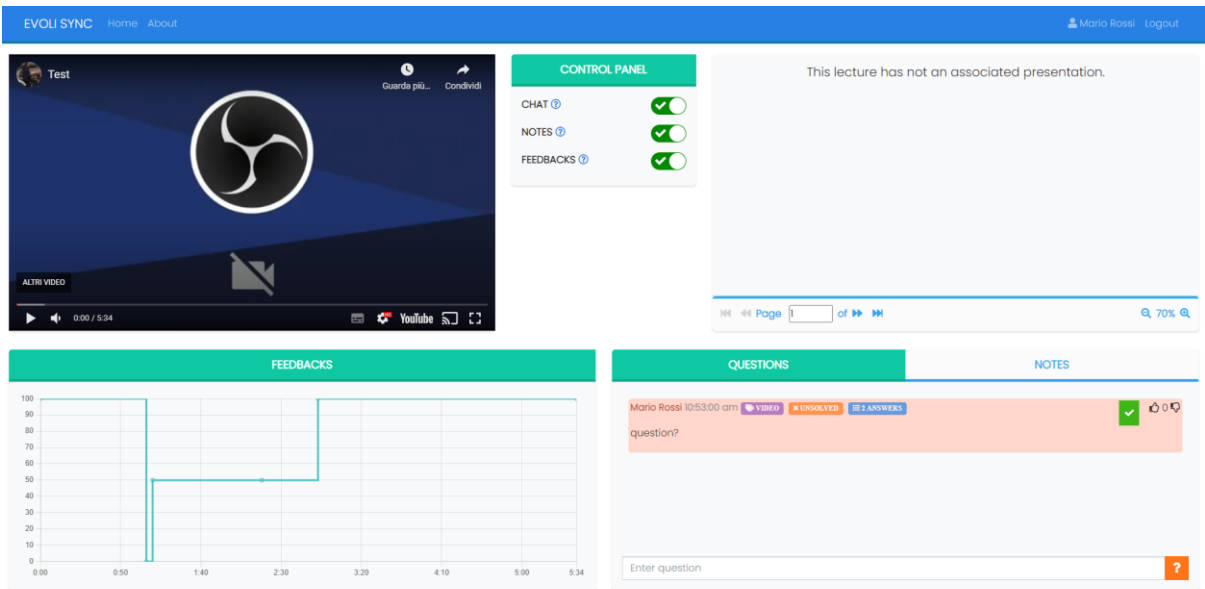


Figure 3.27: Teacher report page

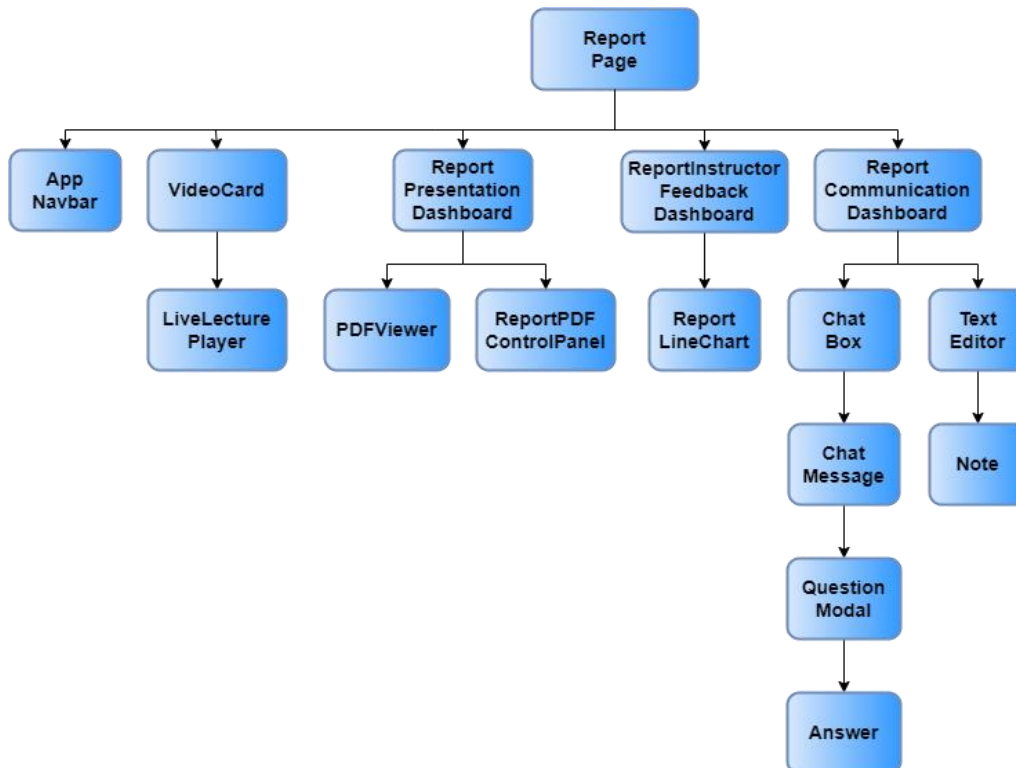


Figure 3.28: Teacher report panel page components

UNAUTHORIZED ERROR PAGE

The page is displayed when a user tries to access a resource without having its authorization, that is, when he tries to access an internal page without being logged in.

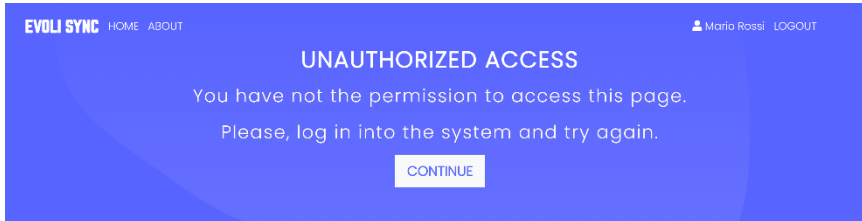


Figure 3.29: Unauthorized Error page

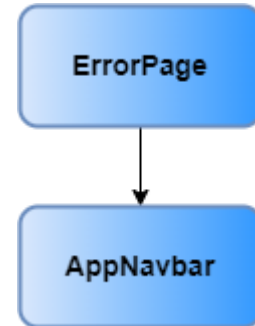


Figure 3.30: Unauthorized Error page components

404 ERROR PAGE

The page is displayed when the user attempts to access a resource that does not exist, triggering the 404 - Not found error .

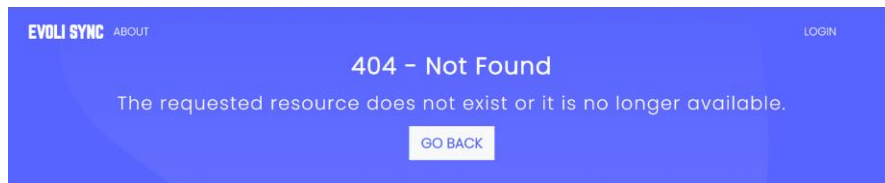


Figure 3.31: 404 error page

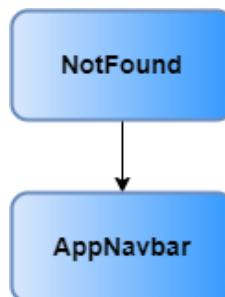


Figure 3.32: 404 error page components

3.6.5. Components

Table 3.11: React components

| Code | Category | Component | Description |
|------|----------|-----------|-------------|
|------|----------|-----------|-------------|

| | | | |
|---|------|------------------|--|
| 1 | Auth | LoginForm | A user can login inserting email and password into this form. If login success, he is redirected in the home page, otherwise an error message is displayed. |
| 2 | Auth | RegistrationForm | A user can register inserting email, first name, last name, password and confirmation password into this form. Once the registration is done, the login form is displayed with the message "You are now registered and can log in!" |
| 3 | Auth | ProfileModal | It permits to the user to edit its profile information: first and last name and eventually the password. |
| 3 | Chat | Answer | It shows an answer to a question message. It is possible to insert like or dislike reaction with the apposite buttons. |
| 4 | Chat | ChatBox | Box containing the chat with the list of chat messages and a form to insert a new question or a new normal message. |
| 5 | Chat | ChatMessage | Chat messages that display differently in case the message is a normal message or a question. If the message is a question it contains a clickable badge to see the answer and a button to solve it. In both cases |

| | | | |
|----|---------|------------------------|--|
| | | | there are the like and dislike reactions. |
| 6 | Chat | QuestionModal | Modal that contain the detail of the question. It shows the question's answers and a form to insert a new answer. |
| 7 | Chat | ReportQuestionBox | Box that contains all the questions associated to the lecture. It is possible to insert a new question also in the report page. |
| 8 | Courses | CourseDetailCard | Simple card that shows the information about the selected course, such as name, instructor and creation date. |
| 9 | Courses | CreateCourseForm | Form able to create a new course inserting its name. |
| 10 | Courses | EnrolledStudents Modal | Modal that show the list and the number of students enrolled to a given course. |
| 11 | Courses | FollowedCourseCard | Card containing the information about a course followed by the user. It contains a button for unenroll from the course. If the card is clicked, then the list of lecture of the course is shown to the user. |
| 12 | Courses | StudentCoursesList | It contains a form to enroll to a new course by inserting its code and the list of courses followed by the student. |

| | | | |
|----|---|----------------------------------|--|
| 13 | Courses | TaughtCourseCard | Card containing the information about a course taught by the user. It contains three buttons: one to copy the course code, one to delete the course and one to show the enrolled students. When the card is clicked the list of lecture of such course is shown to the user. |
| 14 | Courses | TeacherCoursesList | It contains a form to create a new course by inserting its name and the list of courses taught by the user. |
| 16 | Dashboards LiveLecture Dashboards | / Communication Dashboard | Dashboard showing the chat, the notes and the participants. If there is not the presentation, then it displays only the chat and the notes since the participants are displayed in another part of the page. |
| 17 | Dashboards LiveLecture Dashboards | / ControlDashboard | Student control panel that permits to the user to insert understanding reactions. |
| 18 | Dashboards LiveLecture Dashboards | / Feedbacks Dashboard | Dashboard showing the feedback analysis, in particular it show a donut chart and a line chart. If the understanding% goes under a threshold, then a warning is shown. |
| 19 | Dashboards LiveLecture Dashboards | / InstructorControl Dashboard | Control dashboard that shows to the instructor the commands to enable or disable chat, notes and the visibility of the feedback to the students. It also |

| | | | |
|----|---|--|--|
| | | | shows a button the permits to the instructor to end the lecture. |
| 20 | Dashboards LiveLecture Dashboards | / InstructorFeedback Dashboard | Feedback dashboard showing a donut chart an a line chart. Pressing a button it is also possible to see a chart that shows the recap until a certain moment in the lecture. |
| 21 | Dashboards LiveLecture Dashboards | / Instructor PresentationDashboard | Dashboard that shows the presentation viewer and the instructor pdf control panel. |
| 22 | Dashboards LiveLecture Dashboards | / Participants Dashboard | Dashboard that shows the list of the lecture participants. It also show the participants number in a badge. It is shown only when there is no presentation. |
| 23 | Dashboards LiveLecture Dashboards | / Presentation Dashboard | Dashboard that shows the presentation viewer and the pdf control panel. |
| 24 | Dashboards ReportDashboards | / Report Communication Dashboard | Dashboard that shows the lecture questions and the notes associated to the lecture. |
| 25 | Dashboards ReportDashboards | / ReportFeedback Dashboard | Dashboard that show a progress bar indicating the understaning hystory of the current user in the lecture and a line chart that shows the understaning % trend of the students participating to the lecture. |

| | | | |
|----|-------------------------------|---|--|
| 26 | Dashboards / ReportDashboards | ReportInstructor ControlDashboard | Control dashboard that shows to the instructor the commands to enable or disable chat, notes and the visibility of the feedback to the students during the vision of the report. |
| 27 | Dashboards / ReportDashboards | ReportInstructor Feedback Dashboard | Like the student dashboard it show the line chart with the understanding% trend in the lecture but this dashboard is always visible and cannot be disabled. |
| 28 | Dashboards / ReportDashboards | ReportPresentation Dashboard | Dashboard that shows the presentation viewer and the pdf control panel. |
| 29 | Error | Error | Toast container that displays errors in toast elements. |
| 30 | Feedbacks | LectureDoughnut Chart | Donut chart that shown the number of current student connected to the lecture that is understanding 0, 1 and 2. |
| 31 | Feedbacks | LectureLineChart | Line chart that shows the participants' understanding% in the last 5 minutes. |
| 32 | Feedbacks | RecapLineChart | Line chart that show the undestanding % trend from the beginning to lecture until the current moment. |
| 33 | Feedbacks | ReportLineChart | Line chart that shows the understanding% trend of the students that were connected to |

| | | | |
|----|-----------|----------------------------|---|
| | | | the lecture from the start to the end of it. |
| 34 | Feedbacks | Understanding Switch | It contains a slider that permits to the user to insert understanding reaction from a value of 0 to a value of 2. |
| 35 | Feedbacks | Understanding Warning | Animated warning shown when the understanding% goes under a fixed threshold . |
| 38 | Lectures | CreateLecture Modal | Modal that permits to create a new lecture. It requires the lecture name, the scheduled start time, the scheduled end time, the YouTube video url and the presentation path. |
| 39 | Lectures | InstructorLectures Table | Sortable table that shows the list of lecture of the selected course. |
| 40 | Lectures | InstructorLecture TableRow | Table tow that shows the information about a lecture, such as the name, the timetable, the status and the possible actions. The actions are: delete the lecture, update the lecture, go to live lecture, download the slides, download the notes, go to report. |
| 41 | Lectures | StudentLectureTable | Sortable table that shows the list of lecture of the selected course. |

| | | | |
|----|--------------|----------------------------|--|
| 42 | Lectures | StudentLecture TableRow | Table row that shows the information about a lecture, such as the name, the timetable, the status and the possible actions. The actions are: go to live lecture, download slides, download notes, go to report. |
| 43 | Lectures | UpdateLecture Modal | Modal that permits to update the lecture information such as name, timetable, presentation path and YouTube video url . |
| 44 | Navbars | AppNavbar | Application navbar. It contains the links to the current accessible pages, such as about page, home page, index page and also a logout button if the user is logged. |
| 46 | Notes | Note | It shows the detail of the notes, including the notes creator, the text of the note and the possibility to insert like or dislike reactions. |
| 47 | Notes | TextEditor | It shows either a text editor to insert a new note, that can be private or shared, or the notes lists. If there is a presentation it shows the note tagged to the current slide, else it shows all the notes. We use the ckeditor library to embed a rich text editor. |
| 48 | Participants | ParticipantsBox | Box that contain the list of participants. |

| | | | |
|----|--------------|---------------------------|--|
| 49 | Participants | UserCard | Card that show the participant name and surname. It is in a different color depending on the participant is the instructor or a student. |
| 50 | Presentation | PDFControlPanel | Student control panel. It shows the commands to change page, go to teacher page and change zoom. |
| 51 | Presentation | PDFInstructorControlPanel | Instructor control panel. It shows the commands to change page and change zoom. |
| 52 | Presentation | PDFViewer | It shows the presentation slides. We use react-pdf library to implement this functionality. |
| 53 | Presentation | ReportPDFControlPanel | It shows the commands to change page and zoom. |
| 54 | Video | LiveLecturePlayer | Lecture video player. When the student enter in the lecture the video loads and then a clear understanding (value = 2) is sent to the server. When it is unmounted an understanding of -1 is sent. In this way the server can understand when a user connects and disconnect from the lecture. |
| 55 | Video | LiveLectureVideoCard | Card the shows the YouTube video player in a responsive way. |
| 56 | Video | VideoCard | It shows the video to the instructor in a responsive way. |

| | | | |
|----|-------|---------------|---|
| 57 | Video | YouTubePlayer | <p>YouTube video player. It exploits the YouTube IFrame player API to shows a YouTube video and control it.</p> <p>In case the video has an incorrect link, then it shows an error alert.</p> |
|----|-------|---------------|---|

3.6.6. Application global state

Our application state is handled by the React Context API. Context API solve the problem to maintain a global application state solving a lot of problems related to state management and how the states object are passed to the components.

In EVOLI Sync the state is constituted of 4 different contexts:

AUTH CONTEXT

It contains the current user information and the functions to login and logout.

Table 3.12: Auth context

| Property | Description |
|---------------------|---|
| currentUser | The current logged user |
| registerUser | Login the user into the system setting the current user |
| logoutUser | Logout the user, set the current user to null |

PROFILE CONTEXT

Contains the user profile information, such as its followed courses, it taught course, the chosen course and the relative lectures with the operations to manage them.

Table 3.13: Profile context

| Property | Description |
|----------------------|--|
| courses | The list of taught courses by the current user |
| currentCourse | The selected taught course |

| | |
|------------------------------|--|
| followedCourses | The list of followed courses by the current user |
| currentFollowedCourse | The selected followed course |
| lectures | List of lectures of the selected taught course |
| currentLecture | The selected lecture |
| followedLectures | The list of lecture of the selected followed course |
| loading | Loading state during the operation |
| getCourses | Retrieve and set the taught courses by the current user |
| getFollowedCourses | Retrieve and set the followed course by the current user |
| selectCourse | Retrieve and set the current taught course |
| selectFollowedCourse | Retrieve and set the current followed course |
| getLectures | Retrieve and set the lecture of the selected taught course |
| getFollowedLectures | Retrieve and set the lecture of the selected followed course |
| createCourse | Create a new course and add it to the list of taught courses |
| subscribeToCourse | Enroll a student to a course and add it to the list of followed course |
| deleteCourse | Delete a taught course and remove it from the list of taught courses |
| unsubscribeCourse | Unenroll a student from a course and remove it from the list of followed courses |
| deleteLecture | Delete a lecture of a taught course |

| | |
|---------------------------|--|
| createLecture | Create a lecture and add it to the list of the lectures of the current selected course |
| updateLecture | Update lecture information |
| removePresentation | Update a lecture removing its presentation |
| selectLecture | Select the current lecture |

VIDEO CONTEXT

It keeps track of the YouTube video player in order to control it from all the application components. Once the player is ready it saves also the video duration in a context variable.

Table 3.14: Video context

| Property | Description |
|--------------------|---|
| player | Video player reference |
| duration | Video duration |
| setPlayer | Set the video player |
| setDuration | Set the video duration once the video player is ready |

SOCKET CONTEXT

Save the client socket reference in order to receive and send events from all application components. Furthermore, it contains the method to connect the socket with the server. In this context we also establish the initial socket connection. The connect socket method is called each time a user enters in the lecture page or report page. When he leaves the page the connection is disrupted.

Table 3.15: Socket context

| Property | Description |
|---------------|----------------------------|
| socket | Reference to client socket |

| | |
|----------------------|--|
| connectSocket | Connect the socket client to the socket server |
|----------------------|--|

4. Evaluation

This chapter reports on the evaluation of the tool, in terms of usability and perceived usefulness. The evaluation involved 74 higher-education students.

4.1. Evaluation context

EVOLI Sync was tested in a real-life environment, during two lessons of the “Communication & Argumentation” course taught by Professor Nicoletta Di Blas at the Politecnico di Milano. Overall, 74 students took part in the experimentation. The course deals with communication principles borrowed from linguistic sciences, semiotics and argumentation theory. 60 students were on location while 14 were connected from home; not all responded to the questionnaire (which was presented as optional).

4.2. Questionnaire

The questionnaire entailed 27 questions and required from 5 to 10 minutes for completion. After a first section meant to profile the respondents, it probed the perceived usability and usefulness of the tool. Though administered in the context of higher education, the questionnaire is already set for gathering feedback from users in different contexts and school levels.

Table 4.1: Questions

| # | Question |
|---|-----------|
| 1 | Gender |
| 2 | Age range |
| 3 | Level |

| | |
|----|---|
| 4 | Can you please tell us more? (what are you studying) |
| 5 | On which device have you used EVOLI Sync? (More than one option possible) |
| 6 | You used EVOLI Sync remotely or on location? |
| 7 | How clear was the understanding slider? (From 1 to 5) |
| 8 | How clear was the chat message and questions? (From 1 to 5) |
| 9 | How clear was the students notes functionality? (From 1 to 5) |
| 10 | How clear was the doughnut chart? (From 1 to 5) |
| 11 | How clear was the linear chart? (From 1 to 5) |
| 12 | Tick the action(s) you had performed during the lesson? |
| 13 | Do you think that it is useful to see the understanding level of the other students (the doughnut chart)? |
| 14 | Do you think that it is useful to see the trend of understanding of the overall students (the line chart)? |
| 15 | Why? |
| 16 | Do you think it is useful to distinguish the questions from the normal chat messages? |
| 17 | Why? |
| 18 | How much do you agree with the following sentence: I feel more confident in use the understanding slider instead of raise my hand to let the professor know I am not understanding. (From 1 to 5) |
| 19 | Do you think that seeing the other's level of undestanding have influenced your use of the understanding slider? |
| 20 | How useful do you think the possibility to ask question, chat, write notes anonymously? (1=useless, 5=extremely useful) |

| | |
|----|---|
| 21 | How useful do you think the possibility to review the feedback (including questions, notes and messages) that you've inserted during the lesson after it is finished? (1=useless, 5=extremely useful) |
| 22 | Is there any action you would have liked to perform and you couldn't (because EVOLI Sync does not support it)? |
| 23 | How usable is EVOLI Sync? (On a scale from 1 to 5, where 1=not at all and 5=very much) |
| 24 | Would you like to re-use EVOLI Sync in other courses? |
| 25 | From your point of view, which are the strenght points of EVOLI Sync? |
| 26 | From your point of view, which are the weak points of EVOLI Sync? |
| 27 | Overall, did you appreciate EVOLI? (On a scale from 1 to 5, where 1=not at all and 5=very much) |

Question 1: Gender



Figure 4.1: : Question 1

This question was intended to track the gender of the assessment test participants. More than half is male, the others female and a small minority preferred not to specify it.

Question 2: Age range

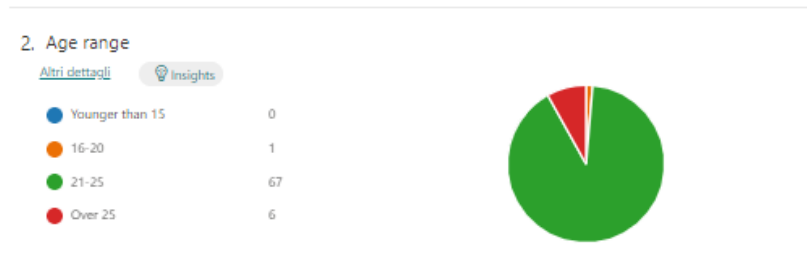


Figure 4.2: Question 2

This question has the purpose to track the age of the participants. It has demonstrated that all the students except one are over 20 years of age with 6 over 25, thus centering the university age range.

Question 3 and Question 4: Level and Can you please tell us more? (What are you studying)



Figure 4.3: : Question 3

All but one of the students confirmed that they were university students. The majority confirm that they are students of computer engineering, management, mathematics, energy, electrical, aeronautics, nuclear, physics, biomedical, geoinformatics or telecommunications while one student studies finance and two design.

Question 5: On which device you used EVOLI Sync?

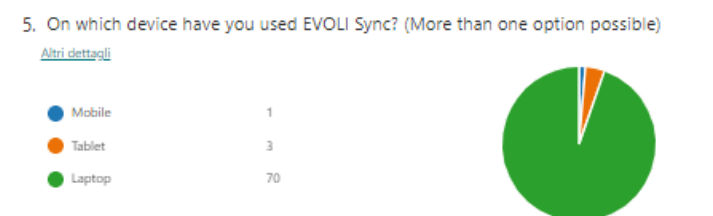


Figure 4.4: : Question 5

With this question, the students stated that they use EVOLI Sync mostly with their laptop, while only a minority use their tablet or mobile phone.

Question 6: You used EVOLI Sync remotely or on location?



Figure 4.5: Question 6

Più di $\frac{3}{4}$ degli studenti hanno affermato di accedere alla piattaforma in presenza, mentre meno di $\frac{1}{4}$ (14 su 74) a distanza.

Question 7, 8, 9, 10, 11: How clear was the understanding slider (7), the chat message and questions (8), the notes (9), the doughnut chart (10), the linear chart (11)?

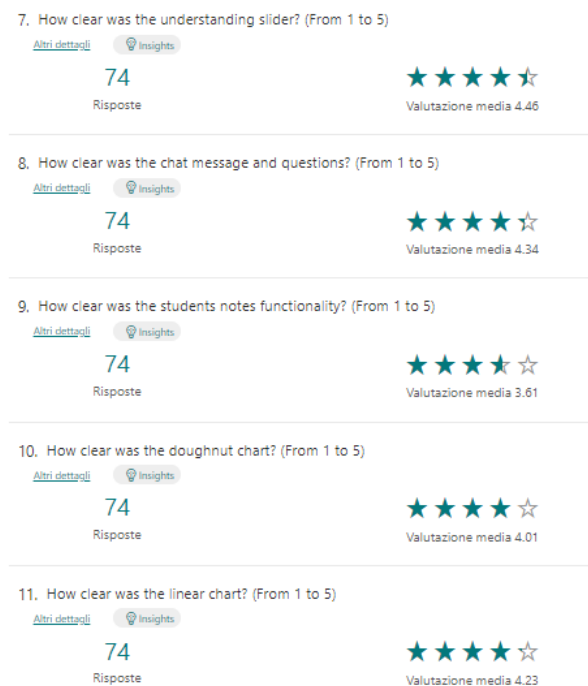


Figure 4.6: Question 7-11

These questions were intended to understand how clear the various features offered to students were found. Overall, the features were deemed clear, with scores ranging from 3.61 (the notes) up to 4.46 (the slider). According to the students it was not clear that the notes were tagged to the single slide and that they were visible only when the tagged slide was viewed. This criticism will certainly be considered in future versions of the application.

Question 12: Tick the action(s) you had performed during the lesson



Figure 4.7: Question 12

This question was intended to track the use of student feedback features. The slider was the most used feature, as expected. Some students have used the chat and entered notes. Probably the number of students using chat and notes will increase as the visual interface is improved. A good part of the students (about 1/3) did not use any functionality.

Question 13, 14, 15: Do you think that it is useful to see the understanding level of the other students (the doughnut chart)? Do you think that it is useful to see the trend of understanding of the overall students (the line chart)? Why?

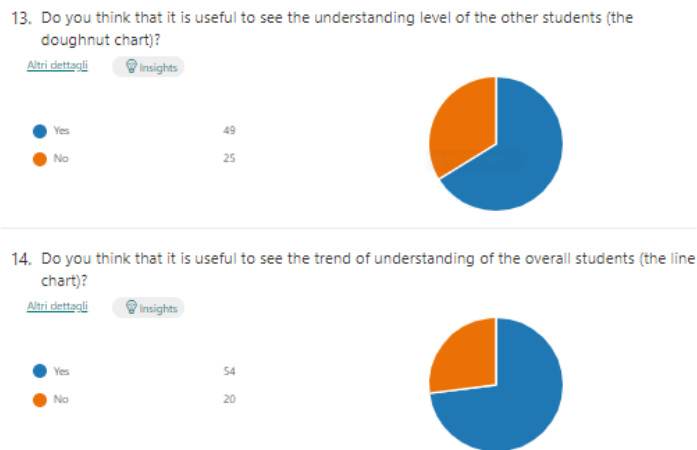


Figure 4.8: Question 13-14

About 2/3 of the students said they found it useful to see the level of understanding of other students, while about 3/4 of the students found it useful to see the trend of understanding of the students (the line graph). The most common reasons were:

“It encourages people to ask more questions, if the general understanding seems to be low.”

“Because in that way, if I feel like I’m not understanding something and I see a lot of people are not understanding either, it is easier for me to not feel scared to ask. Also, if the teacher can see someone is not understanding, he or she may be able to try to be clearer without the student having to tell, and in that way no one is left behind”

“Because it gives a clearer idea as to whether a concept is difficult to me only or if it is really difficult to grasp in general”

Question 16 and 17: Do you think it is useful to distinguish the questions from the normal chat messages? Why?

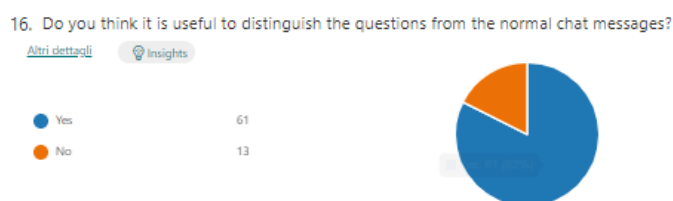


Figure 4.9: Question 16

This question demonstrates that for 61 out of 74 students it is useful to have a distinction between chat messages and questions. This is motivated by the following statements:

“Because it would make it easier to find the questions, which are important to understand (and perhaps even recognize) important concepts explained during the lecture (unlike messages that may contain information which is not really relevant)”

“I think it is important to highlight questions and distinguish from common messages. Indeed the chat may be full of messages of no relevance and some questions may be ignored.”

“Chat messages sometimes are related to check connectivity issues or other out of topic stuff, so maybe it is good to do this differentiation”

Question 18: How much do you agree with the following sentence: I feel more confident in use the understanding slider instead of raise my hand to let the professor know I am not understanding.

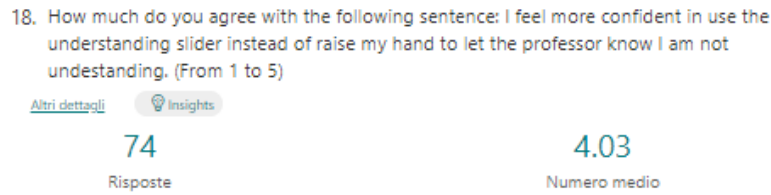


Figure 4.10: Question 18

This question demonstrates that students are much more comfortable using the slider than interrupting the lesson to ask questions. Sometimes, in fact, the student may not have understood to the point of not being able to ask questions and in any case the teacher is able to know at any time the level of general understanding of the class.

Question 19: Do you think that seeing the other's level of understanding have influenced your use of the understanding slider?

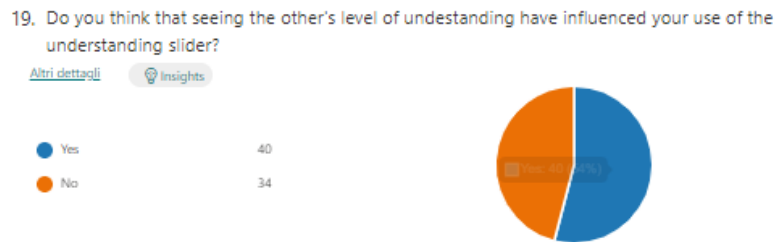


Figure 4.11: Question 19

Questa domanda aveva lo scopo di dimostrare che vedere il grado di comprensione degli altri studenti influenza il proprio comportamento nell'esplicitare il proprio livello di comprensione. In particolare dimostra che sapere che altri studenti non hanno capito, incoraggia il singolo partecipante a mettere in luce le proprie lacune. Infatti, più della metà dei partecipanti ha risposto 'Sì'.

Question 20: How useful do you think the possibility to ask question, chat, write notes anonymously?

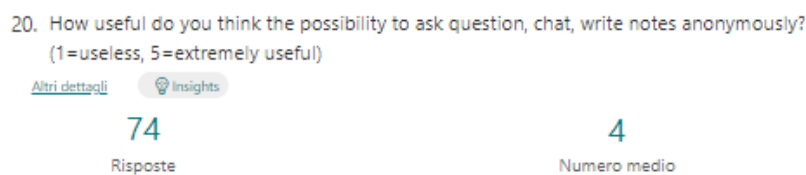


Figure 4.12: Question 20

The students confirmed that they felt it was important to be able to express themselves anonymously. Currently, the summary graphs of students' degrees of understanding do not keep track of who is understanding 0, 1 or 2. Anonymity will certainly be extended to other features in future versions.

Question 21: How useful do you think the possibility to review the feedback (including questions, notes and messages) that you've inserted during the lesson after it is finished?

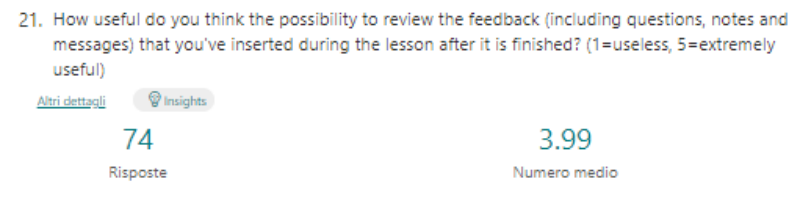


Figure 4.13: Question 21

The students found it useful to be able to review the feedback given even after the lesson was over. In fact, they answered the question with a score of 4 out of 5.

Question 22: Is there any action you would have liked to perform and you couldn't (because EVOLI Sync does not support it)?

Most of the answers were related to the graphical interface. In fact, the students found it necessary to make the chat larger than the video, since the test lessons were mainly focused on the PDF presentation.

“Put the power point presentation in a full screen configuration.”

“I would like to decide the dimensions of “windows”. I mean, in this lecture is not interesting for instance to see the professor because she’s just speaking, I would like to see slides better, but they were quite small.

Also the feedback has not been useful for me, I would have preferred bigger slides, the main thing of my interest.”

“Hide some boxes to give more focus to others.”

Question 23: How usable is EVOLI Sync? (On a scale from 1 to 5, where 1=not at all and 5=very much)

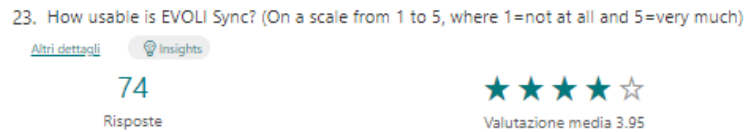


Figure 4.14: Question 23

The students considered EVOLI Sync as a platform with high usability, giving an average score of 3.95 out of 5.

Question 24: Would you like to re-use EVOLI Sync in other courses?

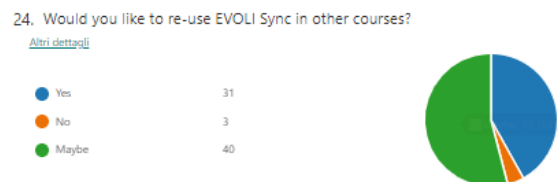


Figure 4.15: Question 24

This question showed that very few students would not re-use EVOLI Sync in other courses, while 31 students would re-use it and 40 would be undecided.

Question 25: From your point of view, which are the strength points of EVOLI Sync?

The majority of students considered the feedback system to be the strong point of EVOLI Sync, which, moreover, is the focus of the thesis. Also liked the fact that the student can go back and forth on the slides regardless of the teacher's position.

“the understanding feedback from students”

“the professor knows if the students are really understanding and the students can say if they’re not understanding without feeling judged”

“The possibility to have the control of the slides or see the understanding level”

“The possibility to indicate your level of understanding and observe that of the whole class, the chat, the possibility of creating notes, the possibility to follow the slides as they are shown from the professor.”

“It is much more interactive than WeBex and it is useful to have live feedbacks. It is nice that closing and reopening the platform the messages do not go away and we still can see it.”

Question 26: From your point of view, which are the weak points of EVOLI Sync?

The weak point was certainly identified in the UI of the lesson and in the poor customization of the screen, as was expected. The most common responses were:

“The possibility to manage interface”

“The distribution of the space (too space given to chat and level of understanding)”

“The impossibility to manage the windows on the page. for example, change the view and only visualize the slides or the professor talking”

Question 27: Overall, did you appreciate EVOLI?

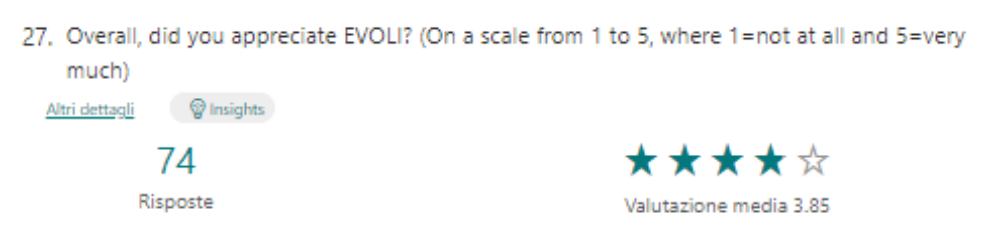


Figure 4.16: Question 27

In general, the platform has shown a good appreciation, with a score of 3.85 out of 5.

4.3. Analysis

CLEARITY. Overall, the students considered the functionalities of the system clear, giving an average score of 4.13 / 5: sliders, messages, questions, notes and graphs. The notes were scored as the least clear functionality (3.61), while the “understanding slider” as the clearest (4.46). Overall, the most used functionality was the understanding slider, while the questions the least used.

UNDERSTANDING ANALISYS. About two-thirds of the students found it helpful to see other students' level of understanding in the graphs, on the ground that this encourages people to ask more questions if their level of understanding is low and, in general, to understand if low personal understanding is a symptom of personal gaps or lack of attention or it is a general class problem, due to the quality of the explanation. It was also noted that knowing if other students also have difficulty on the same topic encourages a participant to interact by asking questions,

reducing the “it’s just me” effect. It can be concluded that having a general idea of the understanding of the audience develops greater group interaction.

The understanding slider was certainly appreciated by the audience who agreed with a vote of 4.03 / 5 to the statement: "I feel more confident in using the understanding slider instead of raising my hand to let the professor know I am not understanding" (of course the hand-raising is meant both in the physical and in the virtual world, i.e. in class and online).

CHAT. 84% of the class found it useful to distinguish normal messages from chat questions. In particular, it is important to identify the questions rapidly since they have more relevance with respect to the normal message, because they represent doubts about the content of the lesson which could be shared among more than one participant.

WEAKNESS. The weak point of the platform was certainly identified in the poor customization of the user screen. In particular, part of the class felt that the graphics takes up too much space and they think that some components are more important than others. In fact, it was deemed necessary to be able to have a greater visual focus on the slides presentation or on the video, overshadowing the other features when not in use. Many students expressed the desire to be able to view the slide panel larger or change the size of the various panels.

CONCLUSION. We can conclude that the tests were generally successful, as only a very few students would not recommend using EVOLI Sync in other courses. The students greatly appreciated the central focus of the project, i.e. the exchange of feedback between students and teachers, and in particular it showed that knowing and being able to compare with the level of learning of other students significantly increases the interactivity of the class group.

5. Conclusion & Future Works

In this thesis we worked on developing an application focused on a video-annotation tool capable of supporting and facilitating communication and the exchange of feedback between the participants of a lesson. We also saw how the application supports two of the three modalities of the HyFlex model.

The questionnaire has yielded quite relevant data on the tool. First of all, it confirms the validity of the approach and the perceived usefulness by its intended users. One of the most important outcome concerns the sharing of the other students' feedback, which seems to be preferred by the majority, and that brings about implications in pedagogical terms that the instructor will need to take into account. The sharing of the feedback creates, in fact, a sense of being "together", almost like an entity that is given the power to speak up, which is exactly one the main objectives EVOLI is after. Furthermore, the questionnaire has provided a set of suggestions to improve the usability of the tool, which needs to become as "transparent" and smooth to use as possible, given the fast-pace of use it is meant to support (during a live class). Among the outcomes, probably the most relevant concerns the size of the graphs and of the widgets and their possible customization. All this is part of the future developments of the tool.

Starting from the results achieved, it is possible to improve the application with future developments:

INTEGRATION WITH EVOLI

Our aim is to expand the tool integrating in EVOLI Sync the functionalities of the previous, asynchronous, annotation tool in order to support both the flipped classroom approach and the synchronous teaching. The harmonized tool would allow a complete and innovative platform: it would be capable of supporting any type of lesson (flipped, in presence, remote and asynchronous) offering a complete video annotation system. In this way EVOLI will be able to fully support the HyFlex model.

PLATFORM ANALYSIS AND IMPROVEMENTS

After the integration between EVOLI and EVOLI Sync, the research will run on two parallel tracks: on one side, the characteristics and pros/cons of the HyFlex teaching will be analyzed through real-life experiments (mainly at POLIMI), on the other, new features will be added to the tool, based on the results of the analysis.

IMPROVEMENT OF THE USER INTERFACE

In every application, a good graphic interface is essential to ensure the best possible User Experience for the user. In particular, the UX derives from three factors:

- Layout
- Functionalities
- Interaction

We want to ensure that the graphic interface of the application is analyzed and possibly modified by a web designer. The goal is to make it easier and more intuitive to use the application. In addition, we want to provide the end user a customization of the graphical interface while viewing the lesson so that the user can choose the interface with which to follow the lesson from different screens, for example giving priority to the video with respect to the slides or vice versa.

INTEGRATION TO VIDEO-CONFERENCING TOOLS

During this thesis, and in particular in the state of the art, we pointed out that, in the world of education, the classic video-conferencing tools such as Google Meet, Cisco Webex Meetings, GoToMeeting, Zoom and Microsoft Teams are used despite none of them. These tools are born with a didactic orientation. We therefore want to propose to build a plugin for the various systems that has the same functionality of EVOLI Sync as a support to the lesson. That is a system that allows you to insert understanding feedback and notes shared between the students of a lesson and, moreover, that is able to provide the teacher (host of the video conference) with an at-a-glance analysis of the progress of student understanding.

NOTIFICATION

It is proposed to introduce a notification system. The user, logged into the platform, will have a notification area in which he will be notified whenever:

- A new lesson is created in one of the courses in which he is enrolled
- You are about to start a lesson in one of the courses in which you are enrolled
- A lesson is changed in one of the courses in which he is enrolled
- Another user answers one of his chat questions
- Other users like or dislike your message, question, answer or note.

AGGREGATE ANALYSIS AND STATISTICS ON SETS OF LESSONS

Currently it is possible to analyze in detail only the single lesson in the appropriate report. We want to propose to create customized dashboards that aggregate and analyze data from a set of lessons, so as to be able to make comparisons between multiple lessons and / or courses.

ANONYMOUS MODE

Currently, all comprehension reactions are anonymous, in the sense that the teacher does not see which students have indicated that they understand 0, 1 or 2. We want to introduce an anonymous mode, in which the user can:

- Messaging in chat anonymously
- Post questions in chat anonymously
- Answering questions anonymously
- Create anonymous notes

VIEWING THE LESSON IN 3D MODE



We intend to propose an alternative way of viewing the lesson, in which, instead of the current panels, the lesson becomes a three-dimensional classroom. The user has a view in the center of the classroom and can move around in this three-dimensional virtual reality. The classroom will contain the elements that are now arranged on the screen, namely: video of the lesson, slides, notes, chat and feedback analysis.

Figure 5.1: Example of 3D classroom model

Bibliography

Carrillo, C., & Flores, M. A. (2020). COVID-19 and teacher education: a literature review of online teaching and learning practices. *European Journal of Teacher Education*, 43(4), 466-487.

Hattie, J. (2009). *Visible Learning: a synthesis of over 800 meta-analyses relating to achievement*. LondonNew York. NY: Routledge.

Hattie, J., & Yates, G.C.R. (2013). *Visible Learning and the science of how we learn*. Abingdon. OX: Routledge.

Hutain, J., & Michinov, N. (2022). Improving student engagement during in-person classes by using functionalities of a digital learning environment. *Computers & Education*, 104496.

Johnson, L., Becker, S. A., Cummins, M., Estrada, V., Freeman, A., & Hall, C. (2016). *NMC horizon report: 2016 higher education edition* (pp. 1-50). The New Media Consortium.

Milman, N., et al., (2020). 7 things you should know about the HyFlex Course Model.

Mischel, L. J. (2019). Watch and learn? Using EDpuzzle to enhance the use of online videos. *Management Teaching Review*, 4(3), 283-289.

Pianta, R. C., Hamre, B. K., & Allen, J. P. (2012). Teacher-student relationships and engagement: Conceptualizing, measuring, and improving the capacity of classroom interactions. In *Handbook of research on student engagement* (pp. 365-386). Springer, Boston, MA.

Shulman, L. S. (2002). Making differences: A table of learning. *Change: The Magazine of Higher Learning*, 34(6), 36-44.

Sigmon, A. J., & Bodek, M. J. (2022). Use of an Online Social Annotation Platform to Enhance a Flipped Organic Chemistry Course.

Singh, R., & Awasthi, S. (2020). Updated comparative analysis on video conferencing platforms-zoom, Google meet, Microsoft Teams, WebEx Teams and GoToMeetings. *EasyChair Preprint*, 4026, 1-9. Webography

- [1] <https://www.evoli.polimi.it/> retrieved online on 24th March 2022
- [2] <https://perusall.com/> retrieved online on 24th March 2022
- [3] <https://ant.umn.edu/> retrieved online on 24th March 2022
- [4] <https://democraticreflection.org/> retrieved online on 24th March 2022
- [5] <https://edpuzzle.com/> retrieved online on 24th March 2022
- [6] <https://videonot.es/> retrieved online on 24th March 2022
- [7] <https://annotate.tv/> retrieved online on 24th March 2022
- [8] <https://media-and-learning.eu/type/tools/make-annotations-on-youtube-videos/> retrieved online on 24th March 2022
- [9] <https://www.engageli.com/> retrieved online on 24th March 2022
- [10] <https://searchsoftwarequality.techtarget.com/definition/3-tier-application> retrieved online on 24th March 2022
- [11] <https://it.wikipedia.org/wiki/Model-view-controller> retrieved online on 24th March 2022
- [12] <https://psicografici.com/jwt-json-web-token/> retrieved online on 24th March 2022
- [13] <https://socket.io/docs/v4/> retrieved online on 24th March 2022
- [14] <https://github.com/remix-run/react-router/blob/main/docs/getting-started/tutorial.md> retrieved online on 24th March 2022

List of Figures

| | |
|--|--|
| Figure 1.1: Democratic Reflection cards | 16 |
| Figure 1.2: EdPuzzleFi | |
| Figure 1.3: VideoNot.esflection cards | Errore. Il segnalibro non è definito. |
| Figure 1.4: EdPuzzle | 17 |
| Figure 1.5: VideoNot.es | |
| Figure 1.6: EdPuzzle ... | Errore. Il segnalibro non è definito. |
| Figure 1.7: VideoNot.es..... | 17 |
| Figure 1.8: VideoNot.es..... | Errore. Il segnalibro non è definito. |
| Figure 1.9: Annotate.tv..... | 18 |
| Figure 2.1: Entity Relationship diagram..... | 26 |
| Figure 2.2: Lesson state diagram | 27 |
| Figure 3.1: 3 level application | 33 |
| Figure 3.2: MVC pattern | 35 |
| Figure 3.3: Components mapping and technologies..... | 36 |
| Figure 3.4: ER diagram..... | 38 |
| Figure 3.5: Logical schema | 41 |
| Figure 3.6: Server side application structure | 42 |
| Figure 3.7: React application structure | 55 |
| Figure 3.8: Page graph..... | 56 |
| Figure 3.9: Index page | 58 |
| Figure 3.10: Register form..... | 58 |
| Figure 3.11: Index page components | 59 |
| Figure 3.12: About page | 59 |
| Figure 3.13: About page components | 59 |
| Figure 3.14: Home page teacher section..... | 60 |

| | |
|---|----|
| Figure 3.15: Lecture modals | 61 |
| Figure 3.16: Home page, teacher's lesson list | 61 |
| Figure 3.17: Home page, student section | 62 |
| Figure 3.18: Home page, student's lesson list | 62 |
| Figure 3.19: Home page components | 63 |
| Figure 3.20: Student lecture page | 64 |
| Figure 3.21: Lecture panels..... | 65 |
| Figure 3.22: Student lecture page components..... | 65 |
| Figure 3.23: Teacher lecture page | 66 |
| Figure 3.24: Teacher lecture page components | 66 |
| Figure 3.25: Students report page..... | 67 |
| Figure 3.26: Students report page components | 67 |
| Figure 3.27: Teacher report page | 68 |
| Figure 3.28: Teacher report page components | 68 |
| Figure 3.29: Unauthorized Error page..... | 69 |
| Figure 3.30: Unauthorized Error page components | 69 |
| Figure 3.31: 404 error page | 69 |
| Figure 3.32: 404 error page components..... | 69 |
| Figure 4.1: : Question 1 | 84 |
| Figure 4.2: Question 2 | 85 |
| Figure 4.3: : Question 3 | 85 |
| Figure 4.4: : Question 5 | 85 |
| Figure 4.5: Question 6 | 86 |
| Figure 4.6: Question 7-11 | 86 |
| Figure 4.7: Question 12 | 87 |
| Figure 4.8: Question 13-14 | 87 |
| Figure 4.9: Question 16 | 88 |
| Figure 4.10: Question 18 | 89 |
| Figure 4.11: Question 19 | 89 |

| | |
|---|----|
| Figure 4.12: Question 20 | 89 |
| Figure 4.13: Question 21 | 90 |
| Figure 4.14: Question 23 | 91 |
| Figure 4.15: Question 24 | 91 |
| Figure 4.16: Question 27 | 92 |
| Figure 5.1: Example of 3D classroom model | 96 |

List of Tables

| | |
|--|----|
| Table 1.1: Comparison between web conferencing tool used in education..... | 19 |
| Table 2.1: Authentication requirements | 27 |
| Table 2.2: Profile requirements | 28 |
| Table 2.3: Live lesson requirements | 29 |
| Table 2.4: Report requirements..... | 30 |
| Table 3.1: User service functions | 42 |
| Table 3.2: Course service functions | 42 |
| Table 3.3: Lecture service functions | 43 |
| Table 3.4: Chat service functions | 44 |
| Table 3.5: Feedback service functions..... | 44 |
| Table 3.6: Express routes..... | 46 |
| Table 3.7: Course routes endpoints | 47 |
| Table 3.8: Lectures route endpoints | 48 |
| Table 3.9: Socket events..... | 51 |
| Table 3.10: Mapping between React routes and pages | 55 |
| Table 3.11: React components | 69 |
| Table 3.12: Auth context | 78 |
| Table 3.13: Profile context..... | 78 |
| Table 3.14: Video context..... | 80 |
| Table 3.15: Socket context..... | 80 |
| Table 4.1: Questions..... | 82 |

Acknowledgments

A conclusione di questo elaborato, desidero ringraziare tutte le persone, senza le quali questo lavoro di tesi non esisterebbe nemmeno.

In primis ringrazio la mia relatrice Nicoletta Di Blas, per avermi seguito e supportato in questi mesi di lavoro e per essere sempre stata un sostegno umano e gentile con cui condividere idee e progetti futuri.

Ringrazio la mia famiglia:

- I miei genitori per avermi supportato moralmente ed economicamente, per l'infinita pazienza, per essere stati un sostegno stabile e non un peso e per aver creduto nelle mie capacità più di quanto ci credessi io. Mamma, so che starai piangendo mentre leggi queste righe, ce l'abbiamo fatta.
- Mia sorella per il sostegno emotivo e i confronti in ogni momento di crisi
- Mio fratello per essere sempre stato il mio punto di riferimento
- Tutto il resto della mia famiglia allargata, semplicemente per essere così come sono

Ringrazio la mia fidanzata Bea per esserci stata dal giorno 0 di questo percorso e per conoscermi meglio di quanto mi conosca io stesso.

Ringrazio i miei amici e colleghi Silvano e Antonio per avere condiviso con me amicizia, risate, gioie e fatiche di questa avventura di 5 anni.