



Bilkent University

Department of Computer Engineering

Senior Design Project

Tutorium

Project Specifications Report

Barış Ogün Yörük

Halil Özgür Demir

Mustafa Çağrı Durgut

Oğuzhan Özçelik

Yusuf Miraç Uyar

Supervisor: Can Alkan

Jury Members: Erhan Dolak, Tağmaç Topal

Progress/Project Specification Report

Oct 17, 2022

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/1

1. Introduction

Have you ever thought that the most efficient way to achieve anything is to communicate directly with someone who has accomplished it before? Well, We did. Tutorium is a platform where you can arrange one-on-one video interviews with people who have excelled in an exam you want to take.

An examination is an inevitable part of the education system and whatever happens to the education system it will require some form of testing. One of the biggest problems with traditional exam preparation methods is that the direct applicability of the education given in such methods to exams is not tested. In the orthodox method, more than one instructor who specialized in only one subject is successful in giving detailed information about their field and they help students to have a deep understanding of these fields, although what students really desire is to familiarize themselves with these fields just to pass the exam. However, Tutorium's instructors who have excelled in exams and have their success registered on Tutorium have both expertise in the fields and also a deeper understanding of the exam and its methods than any other traditional instructor.

So, Einstein once said that if you can't explain it simply, you haven't understood it deeply. Just like this, Tutorium says that if you understand it deeply enough so that you have excelled in an exam, you can explain it very simply than any other instructor. If you think so and want to be included in the system as an instructor or student, Tutorium will soon be your indispensable platform with its fast and scalable servers and a great user experience!

1.1 Description

Tutorium is a platform where users can arrange one-on-one video conferences with people who have excelled in an area and have this achievement registered in the system. Tutorium provides a platform for both students and instructors to find what they want with ease. Instructors who have registered their achievements in the Tutorium system share their availability schedule on their page and anyone who is interested in having a one-on-one video interview with that instructor books an available time slot and a meeting settles. Tutorium provides an all-in-one experience where users don't have to register for the meeting. All of the video meetings are conducted inside the platform.

Tutorium may be used by too many users at the same time. Because of that, we have to approach this innovation as a product performance innovation. Live streaming must be smooth enough to appeal to people. In addition, the application should support mass amounts of simultaneous streaming from different users. Also, we have to think about the user side because, even though Tutorium is a specific field in this market, if users are not comfortable using this product, they may prefer other generic products. So we have to approach this innovation as a service too.

Tutorium is a Platform that aims to reach a massive amount of users with low latent servers. Therefore, the system should be scaled in a very optimized way. We are planning to start with building the video conferencing infrastructure. Currently, some alternatives for that are using a p2p Video service or an API that would help us. Tutorium aims to be more than just a cs491/2 project. Tutorium will be designed for real user interaction and production. We

are aiming to ship Tutorium to the real-life market. Therefore, optimization will be the main focus to compete with other alternatives and access wide communities.

Of course, there are rooms to grow and areas to discuss in Tutorium. One of the main targets is having an incremental approach. Tutorium will initially focus on the university exam market in Turkey but will test its infrastructure and goals. As a result of these trials, we think that Tutorium will find its real target audience and market. Moreover, the base application will only support a few automatic documentation verifications of the tutors. This will be improved over time. Server usage will also be optimized as the user count increases.

How the video infrastructure will be designed is going to be one of the first big questions faced by the project. Peer-to-peer or API methods can be used. Implementing screen recording and interactive screen-sharing features during lessons is one of the issues that will require us to make strategic choices. It is planned to automatically create a slide from the whiteboard content that is used during the session. Yet, the specifics and the methodologies of this functionality are not yet decided.

1.2 Constraints

1.2.1 Implementation Constraints

- The project will be implemented as a website on the Internet.
- Git will be used for version control of the codebase. [1]
- GitHub will be used for cooperatively working on the project's repository. [2]
- CircleCI will be used for CI/CD pipeline so that codebase will be controlled and tested with each commit. [3]
- Asana will be used for task management. [4]
- Google Meet will be used for the team's video meetings. [5]
- The project will use a third-party library to implement video conferencing logic.

1.2.2 Economic Constraints

- Since large numbers of users will use the application, multiple servers will be needed to scale the application. For development purposes, a free or relatively cheap server will be enough. In the case of production, using one of the Cloud Providers might be needed. In that case, the price will be related to the number of users.
- For all the development tools, free versions will be used. In case of an unexpected price, another new tool will be used instead. (i.e., development tools [Asana, CircleCI, Google Meet] mentioned in the implementation constraints are subject to change.)
- We will buy a domain name to publish our project.

1.2.3 Ethical Constraints

- Tutors' information should be viewed without leaking too much of their personal information.
- Documents of tutors are confidential, so they should be securely stored.

- Contents of the meetings should be encrypted so that adversaries cannot watch them.
- The payment system should be safe so that we don't face a leaking problem.

1.2.4 Sustainability Constraints

- The application will be reviewed at regular intervals once it finishes. The purpose of it is to keep technologies and tools used in the project updated, enhance performance, and fix unwanted behaviors and bugs.
- New features regarding education tools might be added after the project finishes.
- In case of an unexpected number of users, the project will have a scalability logic so that the application will keep running without intervention.

1.2.5 Social Constraints

- The project's initial implementation will be in Turkish since our target market is Turkey. We will first consider the exams of Turkey, such as YKS, ALES, and KPSS. The addition of other countries will be done after the project.
- There should be a request & complaint email for solving disputes between students and teachers. For example, teachers can be different people from how they introduced them to the system.

1.2.6 Technological Constraints

- Users will need a microphone and a camera.
- Users need a sufficient internet connection for the best experience since it is needed for both interactive real-time education tools and video conferences.

1.2.7 Time Constraints

- All the basic functionalities of the project must be finished before the deadline, such as login & register logic of students and teachers, verifying exam results, viewing teachers, setting up & canceling meetings, and uploading documents.
- At least video conferencing and interactive whiteboard features must be finished before the deadline.

1.3 Professional and Ethical Issues

- Content publication of tutors should be checked to verify the published content belongs to them.
- There should be an option for users to report any inappropriate content published by the tutors.
- The payment system included in the application must follow each online payment-related law of the countries in which the payment system is implemented.
- Users should be informed and ask for permission for each user's data that is collected and stored.
- Any personal information of the users that are stored should be stored securely.
- There should be an option for users to report others that they are involved with in a lecture session.

1.4 Similar Products and Technologies

There are products like Zoom [6] and Google Meet [5] as video conferencing applications. Also, there are applications like Udemy [7] that provide educational courses via video streaming. Although, neither of those applications provides a private lecture and schedule-oriented system.

Our application differs from applications like Zoom and Google Meet in two main aspects. First, our application is more oriented toward lecture giving. Meaning it will include features like an interactive whiteboard, slide sharing, optimized whiteboard recording and storage, and automated slide creation from the whiteboard at the end of the session.

Secondly, we are planning to conduct video streams to be conducted as a peer for peer in the sake of scalability. We might decide to use an API for video streaming purposes and try to add special functions for educational purposes but our group intends not to do that.

Compared with applications like Udemy, our application differs as it is more focused on private lecturing and automated verification of tutors for their profession. Our application will provide live streaming which will provide live interaction with the lecturer and the student. Most of the features will be focused on this live interaction fundamental.

2. Requirements

2.1 Functional Requirements

- Users can register to the system.
- Users can send documents to the system to be the tutor of a specific subject.
- Verification of the documents will be automated by the system.
- Students can see the expertise of an instructor from their profile.
- Students can reserve times from instructors.
- Students and tutors can meet at peer-to-peer video conferences.
- Users can use educational tools interactively.
- Educational tools such as interactive whiteboards should be saved and used later by the tutor.
- There will be an automated system that will create a slide show from the changing status of the whiteboard throughout the session.
- Tutors can make quizzes after classes using the materials they used in lessons.
- Students can pay for the classes.
- Tutors can get their pay from the payments.

2.2 Non-Functional Requirements

- **Security:** Tutors' documents should be protected. Payments should also be safe.
- **Scalability:** The system should be able to handle 500 concurrent meetings.
- **Extensibility:** The system should be suitable for new features to be added. Document automation will be added for each subject type separately. It should support the addition of new document types for new professions.
- **Usability:** The application should be simple enough to access broad communities.
- **Performance:** Live streaming should not be lagged for healthy conversation.

3. References

- [1] <https://git-scm.com/about> [Accessed: Oct 12, 2022]
- [2] "Let's Build from Here". <https://github.com/about> [Accessed: Oct 12, 2022]
- [3] "Winning businesses are powered by effective, fast-moving software teams". <https://circleci.com/about/> [Accessed: Oct 15, 2022]
- [4] "We're in business to help you thrive". <https://asana.com/company> [Accessed: Oct 15, 2022]
- [5] <https://meet.google.com/> [Accessed: Oct 12, 2022]
- [6] "We deliver happiness". <https://explore.zoom.us/en/about/> [Accessed: Oct 14, 2022]
- [7] "We share knowledge with the world". <https://about.udemy.com/> [Accessed: Oct 14, 2022]