

Master's Programme in Computer, Communication, and Information Sciences Programme

Introducing Singular-Vision and Community-First concepts to design and implement the next generation of TechClass Learning Environment

Farhad Eftekhari

Copyright ©2020 Farhad Eftekhari

Author Farhad Eftekhari

Title of thesis Introducing Singular-Vision and Community-First concepts to design and implement the next generation of TechClass Learning Environment

Programme Master's Programme in Computer, Communication, and Information Sciences

Major Computer Science

Thesis supervisor Prof. Petri Vuorimaa

Thesis advisor(s) Dr. Joobin Gharibshah

Collaborative partner TechClass Oy (Part of Business Finland Tempo Project Grant)

Date 1.12.2020

Number of pages 74 + 6

Language English

Abstract

The Learning Environment is the heart of every Learning Management System since students are the most important stakeholder when it comes to implementing platforms to deliver educational content. Satisfying students, as the most important user-group, requires a significant effort in many diverse fields, which pushes service providers to overtake sizeable research to understand such diverse fields before starting the implementation process, and then aim for a proper and researched direction to assure the effort will get endorsed and welcomed by the students. Nowadays, in each field, students have various choices, which leads to making the industry more competitive than ever. Furthermore, knowing the fact, the students as the users of our platform do not have unlimited time makes service providers more responsible for assuring they are offering the students services, which are reliable, efficient, and enjoyable to use, which in many of the cases, the users are the younger generations. Observing current popular and trending social platforms, we realize that such services' providers considered the users' behavior and requirements essential points. In this thesis, first, I will look for essential fields, which are essential to pave the road to analyze such systems. I will discuss approaches, which can lead to implementing a system, which can be a successful case in reality. Finally, I aim to represent a new Learning Environment as the Future of Learning Environments named "TechClass next-gen Course Space", and propose methods to validate our result.

Keywords e-learning, Virtual Learning Environment, Learning Management Systems, Web Development, Artificial Intelligence, Quality Assurance, User Experience.

Contents

| | |
|--|----|
| Preface | 7 |
| Abbreviations..... | 8 |
| 1 Introduction | 9 |
| 1.1 Distance Learning in the Pre-internet Era | 9 |
| 1.1.1 Correspondence Education | 9 |
| 1.1.2 Multimedia Distance Education..... | 9 |
| 1.1.3 Computer, Internet, and Distance Learning..... | 9 |
| 1.2 Summary of the Introduction | 10 |
| 1.3 Research goals, objectives, and questions..... | 11 |
| 1.4 Structure..... | 11 |
| 1.5 My role in the project..... | 11 |
| 2 Related Work..... | 13 |
| 2.1 Competitors..... | 14 |
| 2.1.1 Google Classroom..... | 14 |
| 2.1.2 Moodle | 14 |
| 2.1.3 Docebo | 14 |
| 2.1.4 Easy LMS | 15 |
| 2.2 Comparison | 15 |
| 2.2.1 Registration and authorization..... | 15 |
| 2.2.2 Course management..... | 15 |
| 2.2.3 Assignment and assessment..... | 17 |
| 2.2.4 Discussions and forums | 17 |
| 2.2.5 Pricing/Licensing..... | 17 |
| 3 Starting Point | 19 |
| 3.1 The evolution of Learning Environments in TechClass..... | 19 |
| 3.1.1 Version 1: Basic Edition..... | 19 |
| 3.1.2 Version 2: Advanced Edition..... | 19 |
| 3.1.3 Version 3: Advanced Edition with Course Space | 20 |
| 3.1.4 Version 4: community-first approach..... | 21 |
| 3.2 Existing Functionalities and Features..... | 22 |
| 3.2.1 Course Data Model..... | 22 |
| 3.2.2 Course Designer | 22 |
| 3.2.3 Course View | 25 |
| 3.2.4 Course Space..... | 26 |

| | | |
|-------|--|----|
| 3.2.5 | Mobile view | 27 |
| 3.3 | Analysis | 27 |
| 3.3.1 | Consistency | 27 |
| 3.3.2 | Business | 28 |
| 3.3.3 | MVP mindset | 28 |
| 3.3.4 | New Features | 28 |
| 3.4 | Design..... | 28 |
| 3.4.1 | Flow | 28 |
| 3.4.2 | Design Systems | 29 |
| 3.5 | Development..... | 30 |
| 3.5.1 | API-first approach..... | 31 |
| 3.5.2 | Writing testable code..... | 32 |
| 3.5.3 | Choosing a modern front-end development solution | 32 |
| 3.5.4 | Performance enhancement | 32 |
| 3.5.5 | Security | 34 |
| 3.5.6 | Handling logs and errors..... | 34 |
| 3.5.7 | Automate the deployment | 36 |
| 3.6 | Quality Assurance..... | 36 |
| 3.6.1 | Improve test coverage | 36 |
| 3.6.2 | Performance and Availability metrics | 36 |
| 3.7 | Data safety solution, GDPR, and Legal Affairs..... | 37 |
| 4 | Proposed Method..... | 38 |
| 4.1 | Development Method | 38 |
| 4.2 | Analyzing the Flow | 39 |
| 4.3 | Modern Platforms with Similar Mindset..... | 39 |
| 4.3.1 | Mixer..... | 39 |
| 4.4 | Data Models..... | 42 |
| 4.5 | Layout and Vision..... | 42 |
| 4.5.1 | Wireframe | 42 |
| 4.6 | Obtaining Feedback..... | 43 |
| 5 | Result | 44 |
| 5.1 | Singular Vision | 44 |
| 5.1.1 | Mock-up..... | 44 |
| 5.1.2 | Final Design | 47 |
| 5.1.3 | Mobile view | 48 |

| | | |
|-------|---|----|
| 5.2 | Fixing the direction..... | 50 |
| 5.3 | Design and Design System..... | 51 |
| 5.3.1 | The characteristics and strength of our design system | 51 |
| 5.3.2 | How we improved UI and UX of the new learning environment | 51 |
| 5.3.3 | Comparison between the old and the new content editor..... | 51 |
| 5.4 | Technical Improvement..... | 53 |
| 5.5 | Quality Assurance Validation..... | 54 |
| 5.6 | Integrations | 57 |
| 5.6.1 | Microsoft Teams..... | 57 |
| 5.7 | GDPR and Legal Affairs..... | 58 |
| 5.8 | Features to Implement..... | 58 |
| 5.8.1 | Discussions | 58 |
| 5.8.2 | Shared Sections..... | 60 |
| 5.8.3 | Learning Paths | 61 |
| 5.9 | Timeline..... | 62 |
| 5.10 | Questionnaire Results | 63 |
| 6 | DISCUSSIONS..... | 67 |
| 6.1 | Limitations | 67 |
| 6.1.1 | Previous learning environment edition limitations | 67 |
| 6.1.2 | Current learning environment limitations..... | 67 |
| 6.2 | New capabilities and upcoming features..... | 68 |
| 6.2.1 | Smart Sandboxes..... | 68 |
| 6.2.2 | Feedback System..... | 69 |
| 6.2.3 | Assignments Improvements..... | 69 |
| 6.2.4 | Blended Courses..... | 69 |
| 6.2.5 | More User-friendly approaches..... | 69 |
| 6.3 | Future Research Suggestions (What is next?)..... | 70 |
| 7 | CONCLUSION | 71 |
| 7.1 | Community-first approach | 71 |
| 7.2 | Limitations and implementation of features..... | 72 |
| 7.2.1 | The decision about having a separate Design System..... | 73 |
| | References | 75 |
| A. | Questionnaire..... | 78 |
| B. | Data Models..... | 79 |

Preface

I want to thank Professor Petri Vuorimaa, for guiding me through my thesis, in addition to advising and supporting me through my studies. Petri enabled me to broaden my vision to achieve more milestones academically and professionally in the field of education. Furthermore, I want to thank my instructor Dr. Joobin Gharbshah for his guidance.

I also want to thank my company employees at TechClass Oy. Their significant efforts made it possible to make our vision into an innovative and robust product, which made the process of learning more enjoyable and practical for thousands of students.

Finally, I am grateful for the given opportunity to study in Finland. I moved to Finland in 2013, and I was able to focus on my goals and dedicate my time and effort to have the impact that I desire. And yet, it is just a beginning.

Helsinki, December 2020
Farhad Eftekhari

Abbreviations

| | |
|---------|--|
| AI | Artificial Intelligence Agency |
| AICC | Aviation Industry CBT Committee |
| API | Application Programming Interface |
| CI/CD | Continuous Integration and Continuous Delivery |
| CLR | Common Language Runtime |
| CSRF | Cross-site Request Forgery |
| DBMS | Database Management System |
| EF | Entity Framework |
| GDPR | General Data Protection Regulation |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| IDE | Integrated Development Environment |
| JSON | JavaScript Object Notation |
| JWT | JSON Web Tokens |
| LMS | Learning Management System |
| ML | Machine Learning |
| MVP | Minimum Viable Product |
| NLP | Natural Language Processing |
| QA | Quality Assurance |
| SCORM | Sharable Content Object Reference Model |
| SLA | Service-level Agreement |
| SSO | Single Sign-on |
| UAT | User Acceptance Testing |
| URL | Uniform Resource Locator |
| WYSIWYG | What You See Is What You Get |

1 Introduction

Nowadays, learning is not limited to classrooms in schools, universities, and educational centers. Anyone's smartphone, computer, and laptop can easily turn into a learning environment.

As Al Odeh-Haifaa Elayyan states in his book *Toward quality assurance and excellence in higher education*, "The term "e-learning" has only been in existence since 1999 when the word was first utilized at a CBT system's seminar." Later, virtual learning and distance learning were also developed to lexicalize the same meaning. [1]

1.1 Distance Learning in the Pre-internet Era

One may think distance learning dates back to the invention of the internet, but even before the entrance of computers and the internet into our lives, people enjoyed distance learning.

In her article "Serving the System: a critical history of distance education," Jennifer Sumner talks about distance education's critical history. In her article, three generations of distance education are introduced [2]:

"In principle, the introduction is like the abstract, only broader in scope and more detailed. The introduction generally describes the following:

- The First Generation: correspondence study
- The Second Generation: multimedia distance education
- The Third Generation: computer-mediated distance education"

1.1.1 Correspondence Education

Printed materials and postal service were two basic requirements for correspondence study, and Isaac Pitman's shorthand course is a clear example of correspondence education.

In 1840, in England, Isaac Pitman offered the first recognized correspondence courses in shorthand (Verduin & Clark, 1991) to aid business administration. [2, p. 273]

In 1873, Anna Eliot Ticknor, an early pioneer of American correspondence education, founded the Society to Encourage Study at Home, with a mostly female clientele (Holmberg, 1986), in step with an era slowly began to support women's education.

In her book, she also mentions that by the end of the 19th century, some Canadian, American, and European universities offered distance education. [2, p. 274]

1.1.2 Multimedia Distance Education

This second generation of distance education integrated print with broadcast media, cassettes, and—to some degree—computers (Nipper, 1989). [2, p. 276]

The early multimedia course model of the Open University involved the use of one-way technologies—radio and television broadcasts, and audio and video cassettes. [2, p. 277]

The vast majority of distance education worldwide at the end of the 1980s was still mostly print-based, amid the accelerating growth of modern educational technologies. [2, p. 272]

1.1.3 Computer, Internet, and Distance Learning

PLATO stands for Programmed Logic for Automatic Teaching Operations, a project at the University of Illinois, supported by the US army navy and Air Force at different times from the 1960s to the 1980s. The first PLATO system, produced in 1961, was a single-station, computer-based teaching machine, with pre-programmed feedback or diagnostic capabilities. Subsequent PLATO models would eventually be mass-marketed to schools through the Control Data Corporation's big business tactics during the mid-1980s.

The introduction of IBM personal computing in 1981, a significant milestone in the computing field, placed computing power in millions of people's hands, and the internet became publicly available on 6 August 1991. Since then, these two have contributed to noticeable changes in different aspects of people's lives, including education, and revolutionized eLearning.

1.2 Summary of the Introduction

The eLearning portals were an excellent resource for the education systems. By removing the need to have teachers and instructional staff present in every classroom, the program unlimitedly duplicates their teaching capacity. No matter how many students and classrooms the system has, it can serve those students by providing dynamic eLearning solutions once an appropriate eLearning portal is developed. [3]

In the education sector, like any other industry, technology allows us to look for innovative ways to implement new possibilities and, meanwhile, save costs. The critical problem is finding ways to find a solution that provides us with new resources and keeps us consistent with the underlying reason we provide these services.

In TechClass, for near six years, we provided generations of our learning environments. First, I am going to discuss their benefits and flaws.

eLearning is becoming an essential part of university learning management systems. Many educational centers around the world are using eLearning to enhance their traditional learning system. The eLearning Action Plan of the European Commission defines it as the “use of new multimedia technologies and the internet to improve the quality of learning by facilitating access to resources and services as well as remote exchanges and collaboration [4].

A Learning Management System (LMS) is software designed specifically to create, organize, distribute, manage, and deliver educational courses and training programs. LMS is born from eLearning and essentially consists of two major components:

- A core functional platform to upload structured content in texts, images, PDFs, slideshows, evaluative tests, certificates, etc., to manage them as required.
- A user interface is accessed by the lecturers, administrators, and students.

LMSs as a provider of functions and tools such as course management, group chats, and online discussions, and documents [5] in the education sector is used for employee training and orientation by businesses, organizations, government agencies, or offering education by academic institutions. Some key features include a user-friendly design, multi-domain access, content interoperability, etc., leading to an automated, fast, cost-effective, and trackable learning process.

There has been an increase in the number of eLearning systems, especially LMS's, which provide organizations and institutions with a variety of options to choose from. It is essential

to compare different LMS's to analyze their effectiveness, the features that make them unique, and their strengths and weaknesses [6].

Among many LMS's in the market, TechClass' is a special one that provides a community-first approach as a practical and innovative solution regarding common eLearning portal problems. TechClass enables lecturers to manage their teaching activities easily and supports students in their learning process as a smart teaching assistant.

This thesis is dedicated to a comprehensive comparison of TechClass's learning environment to other LMS's. The choice of comparands has been subjective and among the most popular ones in the market, among the open-source systems and those with license fees. The thesis will then discuss the current edition of TechClass Learning Environment and the challenges, analysis, design, and implementation of the next Learning Environment of the Tech-Class platform.

1.3 Research goals, objectives, and questions

The goal of this research is to analyze and implement new capabilities to enhance the experience of learning. I need to obtain ideas from various modern and popular platforms and find ways to migrate them to our services.

Here are the major questions I need to ask to improve the related services:

- What are the limitations of the current system and the expertise and experiences we obtained from the current learning environment to implement the next generation's learning environment?
- What are the new capabilities, which I believe will add significant value to our platform to improve the experience of learning?

1.4 Structure

Chapter 1 delivered an introduction to the learning environments and also discussed the thesis objectives and goals.

Chapter 2 reviews the learning environment of some of the competitors and analyzing their key functionalities.

Chapter 3 explains the previous generations of the learning environment, their key features, and their limitations.

Chapter 4 will demonstrate the development method we used to implement the new generation of the learning environment, analyze the flow and the changes we needed to provide, discuss the similar approaches in the industry, and finally, the method we use to receive feedback from targeted users.

Chapter 5 will deliver the result and improvements we implemented to have a significant step forward to add value to the platform.

Chapter 6 will discuss the upcoming features as a path which the new framework enabled us to pursue. However, more importantly, discuss thesis goals and objectives.

Finally, chapter 7 will deliver a conclusion to the work which I carried out.

1.5 My role in the project

As the CEO of TechClass, I have various teams to manage. A primary goal of such teams is to maintain and improve TechClass Learning Environment. I analyzed, designed, developed, and maintained the first three generations of the TechClass learning environment. Then, I established various teams to re-implement the learning environment with lots of new and improved possibilities.

For the fourth generation, which we call TechClass Smart Online Learning Community, I oversee almost the entire operation. Here are some of the significant roles and responsibilities that I have throughout such a project:

- Brainstorming the ideas and introduce new features
- Analyzing the features and data models
- Consulting development matters with the dev team
- Planning and Recruitment of the members and other resources
- Financial management of the projects
- Finalizing and approving the design decisions and mock-ups
- Planning quality assurance and various test scenarios
- Planning and managing the cloud services and DevOps operations
- Leading the AI team to make the platform smarter and automate more tasks
- Supporting the platform users
- Leading various teams and operating scrum and agile methodologies
- Documenting the progress

2 Related Work

There are several factors to consider for the comparison of different LMS's. There are certain features a standard LMS should have to meet both teachers' and students' needs in the best possible way. Al-Ajlan presented the following criterion to be used for the LMS comparison in his study [7]:

1. Learner tools
 - 1.1. Communication tools
 - 1.2. Productivity tools
 - 1.3. Student involvement
2. Support tools
 - 2.1. Administration tools
 - 2.2. Course Delivery tools
 - 2.3. Curriculum design
3. Technical specification
 - 3.1. Hardware/Software
 - 3.2. Pricing/Licensing

Kavus and Zabadi, has presented another comparison criterion inspired by Al-Ajlan's work based on the five following factors [8]:

1. Whiteboard/Video services
2. Discussion forum
3. File exchange/Internal mail
4. Online journal mail
5. Real-time chat

Ülker and Yılmaz analyzed different properties of LMS's from a brand-independent point of view and proposed the comparison of the following modules for each subject [9]:

1. Course module
2. User module
3. Exam module
4. Assignment module
5. Reporting module
6. Other characteristics (e.g., live chat, forum, polls, etc.)

In this study, I will use various factors the previous researchers used to compare TechClass to the other LMS's. The criterion we propose only focuses on the user experience in learning environments and does not include the other technical aspects:

1. Registration and authorization
2. Course management
3. Assignments and assessments
4. Progress review

5. Discussions and forums
6. Pricing/Licensing
7. Artificial intelligence

2.1 Competitors

Before selecting the right factors, there is an important decision: should the comparands be centrally developed systems with license fees or open-source systems with volunteer developer audiences and without any need for license fees? Although IT managers worldwide seem to split into two in the dilemma of open-source code and major commercial development, this decision is related to capabilities, budgets, and priorities of organizations that employ the LMS rather than personal perspectives. Since institutions are required to pay a service fee to use TechClass, it is fair to include LMS's with license fees in the comparison. To this end, I will use the following LMS's as the comparands:

2.1.1 Google Classroom

Google Classroom is a management framework for cloud-based learning that is part of Google Apps for Education. It is suitable for educational institutions. Students can access Google Classroom from computers, tablets, and smartphones. To manage assignments, import YouTube images, and attach files from Google Drive, users can create Google Docs. It allows users to provide feedback by submitting their comments. To share learning materials for downloading and viewing, teachers can create online classes. To keep track of the progress of students, teachers can create online assignments. Students can quickly apply their assignments via the interconnectivity of Google products such as Google Docs, Google Spreadsheets, and Google Slides. They can access Google Classroom via smartphone applications that are available for iOS and Android. To maximize the performance of their LMS, Google has sufficient infrastructure. [10]

2.1.2 Moodle

Moodle stands for Modular Object-Oriented Dynamic Learning Environment and is a widely known and open-source LMS. It includes comprehensive guides on how to set up your own LMS, tips on how to build and teach Moodle online training courses, as well as a broad group of Moodle users who communicate on different topics. Most notably, it is free of charge and comes with a smartphone application. [11]

2.1.3 Docebo

Docebo is a highly acclaimed LMS endorsed by reputable companies such as Thomson Reuters, Bloomberg, and Sharp. It is a cloud-based LMS solution to provide training programs for employees, partners, and customers. Based on the company's needs, Docebo's various features and applications can be allowed or disabled. Through the app, users can deliver content, manage training, track certifications, and reward learners. Privacy settings allow users to grant special permissions to various clients while protecting them. It offers

unlimited data, classes, administrators, and bandwidth. The framework has also been integrated with third-party software. [12]

2.1.4 Easy LMS

Easy LMS is a cloud-based LMS that is suitable for businesses of all sizes. Users can create lessons and courses that incorporate interactive components such as quizzes, assessments, photos, and videos. This includes designing courses, creating exams, creating quizzes, adding content elements, embedding courses, and branding on a website. Easy LMS allows users to build an infinite number of courses. Some of its advantages include self-registration, the ability to move between different devices (desktop to laptop to tablet), self-paced learning, and the ability to incorporate the courses into students' network. [13]

2.2 Comparison

2.2.1 Registration and authorization

In contrast to TechClass, which requires the student to be affiliated with a partner institution, any person with a Google account can register to Google Classroom and create/attend courses. Each course on the Google Classroom has its link and class code. The class code is a unique 7-digits code, which facilitates the authorization; students can attend the class if they visit the class link and enter the correct class code.

To use Moodle, first, the organization (i.e., the owner) should download and install Moodle on its servers and then configure it. Then, it can set up the platform, create categories and courses, and add and manage users. Each user in Moodle (teacher, student, manager, parent, privacy officer, other) has its account with a username and password.

A student can access a course on its dashboard if and only if a teacher or the manager has added him/her to the list of students of that course.

Like TechClass, first, an organization should pay the licensing fee of the Docebo, and then it can add different types of users. There is a useful feature in Docebo called User Management, which allows the account owner to easily add new users and manage different user roles existing in the platform. A user can be a student, employee, attendee, etc. Each user has its unique username and password to be able to login to the platform. A user can be assigned to a different branch, category, or a specific team of members to facilitate management and collaboration.

After the lecturer creates a course, he/she can send out the invite link of that course to the students so that they can sign up to the platform and enroll in the course. This is very similar to how TechClass works in terms of registration and authentication. The lecturer can write a “Terms and Conditions” text to inform the users before enrolling in the course.

2.2.2 Course management

Google Classroom’s learning environment is similar to a blog where each classwork is analogous to a blog post. TechClass, on the other hand, has designed its learning environment in a more usual fashion, where each course consists of several chapters and sections, and each section intends to teach a specific topic. Although students can change between sections

using the progress bar on the left side of the screen in TechClass LMS, it is impossible to access the next section without finishing the previous one. However, this is not the case in Google Classroom, where the only option is to schedule classwork to be posted in the future.

There are four types of classwork in Google Classroom, assignment, quiz assignment, question, and material. TechClass offers the same as the type of the sections and also the “project.” A notable difference between Google Classroom and TechClass learning environment is that, unlike TechClass, it is not possible to have non-textual content (e.g., image, video, code, etc.) in between the text in classwork of Google Classroom; the non-textual material is attached to a raw text and kept separate from it. In TechClass, the lecturer can add any form of non-textual content between different parts of the text. The supported types of non-textual content are image, video, code, table, attachment file, and video media from YouTube and Vimeo. On the other hand, in Google Classroom, the supported attachments are links, YouTube videos, files (from Google Drive or local drive), Google Docs, Google Slides, Google Sheets, Google Drawings, and Google Forms. Finally, both TechClass and Google Classroom allow the lecturer to send announcements to each course's students.

Moodle does an excellent job of providing easy ways for lecturers to present different materials to their students. Moodle is similar to TechClass in terms of the course's basic structure, albeit with a slight difference that each Moodle course belongs to a category, which is not the case for TechClass. Each Moodle course can have its specific design, dashboard, and structure. Typically, it consists of several topics where each topic contains several activities or resources. Moodle's activity is analogous to a specific section type in TechClass (e.g., content, quiz, assignment, etc.). At the same time, a resource can be a book, file, folder, IMS content package, label, page, and URL. Moodle allows the lecturer to add 15 different types of activities. The main difference between Moodle and TechClass is that Moodle provides many features, settings, and options to customize a course. At the same time, TechClass has focused on a less-complicated fixed-structure and minimalist approach. This difference can be considered as both a pro and a con for Moodle. Although Moodle provides many functionalities to design the course dashboard and its learning environment, its pages are filled with various buttons and options, making it complicated and less user-friendly.

Docebo allows the account owner to create several courses for the target users and design them accordingly. By design, we mean the landing page and structure of the course profile, and the main content. Each course can have as many as possible “Training Material.” Training material can be a new folder, SCORM (Sharable Content Object Reference Model), AICC (Aviation Industry CBT Committee), xAPI, video (or record screen), file, assignment, slide converter, HTML page, survey, and test. Teaching content is most likely provided in either HTML pages or videos. Docebo's user-friendly interface allows course designers to manage content for certification testing and retraining effortlessly. The Docebo LMS Online Training Software can be utilized for selling courses online. Docebo is an ardent supporter of the importance of gamification in learner engagement. Badges, awards, and rankings are applied to increase involvement and course success. This feature is missing in TechClass. Furthermore, Docebo can be utilized together with various video conferencing tools such as Adobe Connect.

Docebo's API enables users to easily integrate third-party applications into the LMS without any additional hassle.

Easy LMS learning environment is different from that of TechClass, Google Classroom, and Docebo. The teaching material is presented via several slides. The content of each slide can be modified by the lecturer using a text editor, which has limited support for multimedia

content. The general style of the slides can also be customized to have a better graphical form.

2.2.3 Assignment and assessment

The format of assignments is mostly the same in both TechClass and Google Classrooms. However, there are some differences, as well. One such difference is that Google Classroom relies on Google Forms to create quiz assignments, while TechClass has its quiz maker tool. Another difference is that the lecturer can add an assignment (quiz, exercise, project, etc.) in a teaching section of TechClass and does not need to create a separate section. This is not possible in Google Classroom, where each classwork is separate and independent from the others. The grading scheme is very similar in both LMS's; quiz assignments are automatically graded, and the other assignments are graded manually by the lecturer.

Moodle's assignments are very interactive. Lecturers can customize the format and design of assignments as much as they want. For example, unlike TechClass and Google Classroom, where each quiz is a set of questions with multiple-choice options, Moodle allows the lecturer to design a quiz section filled with text, image, and other elements, and also place questions in the forms of radio buttons, drop-down lists, drag, and drop, etc., in between them. This enables the lecturers to be creative with the assignments and significantly help students be assessed in a more fun and informative way.

Easy LMS is slightly different from the other LMS's in terms of how it provides the assessment option. In Easy LMS, an "Exam" module represents all types of assessments, such as assignment, exercise, quiz, question, etc. Each exam contains a set of questions, which can be created by the lecturer or even imported from another source, such as a question bank. There are seven possible forms for a question: multiple choice, fill in the blank, free text, small images, medium images, large images, and video/audio.

2.2.4 Discussions and forums

Students can comment on the main page of the course in Google Classroom only if the lecturer has enabled their access. In this way, any person (student and lecturer) can share their opinions and discuss the course. In TechClass, students can chat with the lecturer via an exclusive online conversation tool. Furthermore, they can start discussions on any course section to ask for extra material and explanation and information about errors in the content.

Moodle has a platform for discussion that is very active. There are four basic forum styles in the Moodle culture that help students and educators share ideas by posting comments and organizing workshops. This is an excellent feature that empowers the community by increasing the collaboration of different users.

2.2.5 Pricing/Licensing

Google Classroom is entirely free of charge. Moodle is an open-source LMS, hence free of charge as well.

Docebo's pricing plans are designed to be scalable and accommodate the growing learning needs of enterprise companies. Growth is ideal to launch eLearning projects for up to

300 users quickly. The "Enterprise" plan is designed for high-performing organizations looking for an advanced eLearning solution, as it comes with local cloud hosting, SSO, top-industry SLA, and superior onboarding/implementation.

Easy LMS offers four different pricing plans, where the expensive ones have more features of the platform. A 7-day trial is also provided. Table 1 presents the pricing of the mentioned competitors.

Table 1: Pricing of the mentioned competitors

| LMS | Price |
|------------------|---|
| Google Classroom | Free to begin [14] |
| Moodle | Free on self-hosted plans [15] |
| Docebo | Starting from 1,600\$ per month [16] |
| Easy LMS | €45, €89, and €235 per month based on the plan [17] |

3 Starting Point

In this section, I evaluate the significant aspects of the learning environment to understand the starting points and to which direction I must aim to improve and provide a more convenient learning experience.

3.1 The evolution of Learning Environments in TechClass

3.1.1 Version 1: Basic Edition

To allow previous students who passed the course enrollments earlier to take advantage of the updated material and store the tutorial videos for those who missed a workshop session, TechClass introduced the most primitive version of the basic portal. Each course had a unique course key in the basic edition that allowed the students to get to the course page to access the material and watch the tutorial videos. Figure 1 shows a screenshot from the first generation of TechClass learning environments.

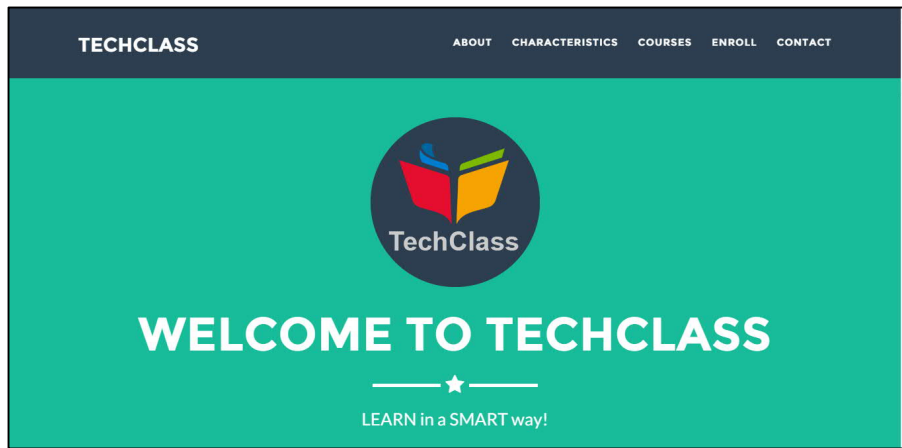


Figure 1: 1st Generation of TechClass learning environments

3.1.2 Version 2: Advanced Edition

TechClass adopted and launched the advanced edition of the platform version. Significant enhancements were made, such as user profiles, enrollment management, enhanced user interface, and an admin portal.

Students were able to register and create a personalized account on the Students portal, enter their personal, email, and university information, read portal notifications, send messages to the lecturer, enroll in available courses, download the course content, submit their assignments, take quizzes for enrolled courses, and finally submit their assessment requests to their lecturers. Figure 2 illustrates a screenshot from the second generation of TechClass learning environments. [18]

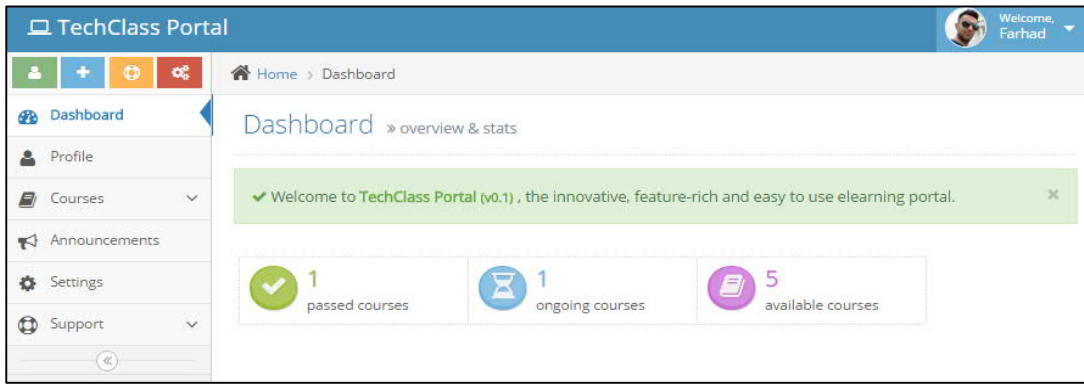


Figure 2: 2nd Generation of TechClass learning environments

3.1.3 Version 3: Advanced Edition with Course Space

The students no longer had to submit their assignments via email by adding Advanced Edition Course Space features. Using the latest released features, they could download the material and submit their assignments and projects.

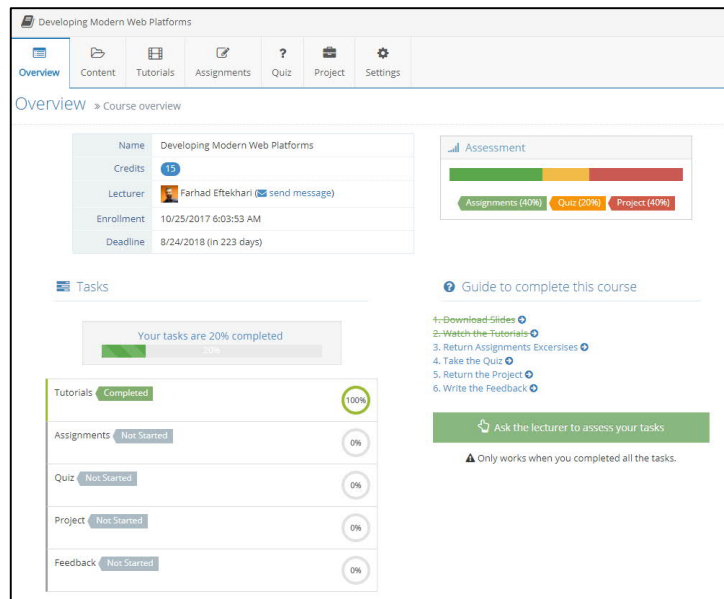


Figure 3: 3rd Generation of TechClass learning environments

With this version of the learning environment, there were several serious problems. First, the teaching environment was not versatile, depriving the lecturers of being creative in offering their materials. Second, it is evident that many students could not facilitate the platform at their best. A weak-structured environment for learning can serve as a misleading resource for education. Third, the portal's innovations were obsolete, and TechClass needed to use the latest technologies to make it easier for new developers to join. Fourth, it was difficult, as a private API, to connect with external services, which is a bad design function for a system due to a lack of integration possibilities. Finally, the students were not aware of each other. There was, therefore, no hope that they would benefit from each other. [19]

Figure 4 demonstrates a list of problems with this version of the learning environment.

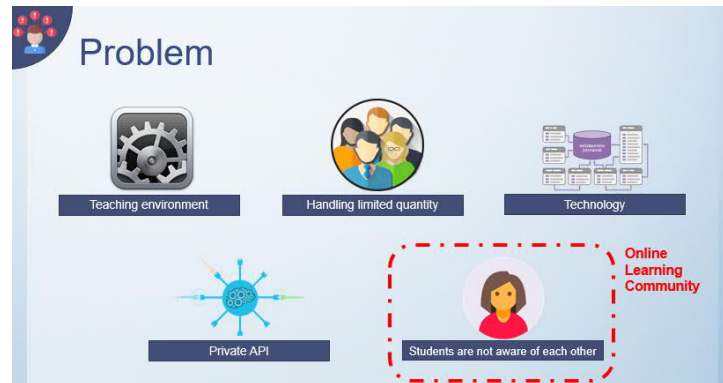


Figure 4: 2nd Generation of TechClass learning environments

3.1.4 Version 4: community-first approach

To switch from an eLearning platform to a Smart Online Learning Community, TechClass needed to introduce a framework that would be powerful and easy to scale. TechClass selected new and updated technologies to allow the development team to develop their code for a long time without rewriting them soon. TechClass placed a great deal of work into evaluating the organizations, assets, and functionality required to reduce the changes needed and manage future functionality to build a scalable framework.

Technology: TechClass used state-of-the-art technologies in each segment of the production when introducing the new system.

Infrastructures: The latest TechClass platform is using Microsoft Azure as one of the most stable, reliable, and expandable cloud services to test, deploy, and host. For development, staging, and production purposes, TechClass utilized various environments using similar resources explicitly specified for such environments.

At its heart, Azure is a public cloud computing platform that can be used for services such as analytics, virtual computing, storage, networking, and much more, with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). It can be used to replace the on-premise servers or to complement them. [20]

Back-end and database: TechClass used C#7.3 and .NET core for back-end implementation. The data access layer is EF Core and Dapper, and TechClass used the repository pattern and Unit of work. TechClass also used EF Migration to keep updating its database.

Controllers are the logic providers for the new RESTful API. For request and response patterns, TechClass adopted the JsonAPI standard. JSON API is a specification for how a client demands to change or retrieve the resources and how a server can respond to those requests [21].

For all API inputs, data validation is performed automatically via the Fluent validation library.

Using Azure DevOps CI/CD, TechClass deployed the back-end project to the Dapper image and ran it in the Azure App service. A CI/CD pipeline automates software delivery process. The pipeline builds code, runs tests (CI), and safely deploys a new version of the

application (CD). Automated pipelines remove manual errors, provide standardized feedback loops to developers, and enable fast product iterations. [22]

Redis used between the docker container as a cache server and SQL Server 2016 as DBMS. TechClass uses Azure Object Storage for image and attachment storage.

Docker is open application development, shipping, and running platform. In order to deploy apps quickly, Docker helps one to isolate applications from infrastructure. One can handle infrastructure with Docker in the same way as to manage software. One can greatly reduce the time between writing code and running it in production by taking advantage of Docker's methodologies for shipping, checking, and deploying code quickly. [23]

Front-end: TechClass used ES6 and React to implement the front-end project of the solution. To monitor URLs, store, and handle side effects, TechClass used react-router, redux, and redux-saga. With saga, all calls to the API are async.

The platform is entirely responsive, and users can use the platform on different devices with a screen size of more than 5 inches, which have browsers and internet connectivity.

User Interface: The most critical aspect TechClass considers in its projects is accessibility and versatility in the new generation of the TechClass platform as a Modern Learning Environment. The most effective Learning Platform is the one that tries to get the students to concentrate on the content, which is what matters most to them. The whole interface will serve as a real learning facilitator to display the essential data in one look. TechClass believed in its platform that the learning hierarchy should be transparent and pursued consistency and predictability, and made the user feel in control. TechClass chose to use Google's material design rules with modifications to adopt specific rules and components to the platform and the environment.

We also used Figma to build mock-ups for the platform. Figma is an interface design application that runs in the browser. [24]

3.2 Existing Functionalities and Features

3.2.1 Course Data Model

Part of the core of every working application and platform is its Entities and Models. Understanding how the models shape the data structure enables us to extend the functional and non-functional requirements of any platform.

Course Models is one of the critical models of the data structure. At the beginning of implementing the system, we aimed to define a light and necessary model rather than have a massive and complicated model. The course model also has various relations with other critical models, such as the User model.

3.2.2 Course Designer

Course Designer is a powerful tool to make it as flexible as possible for the lecturers to design and insert their course material into various chapters and sections and include their assignments and quizzes. Course Designer has a powerful Content Editor to format the content and insert necessary objects. Figures 5, 6, and 7 represent various screenshots from Course Designer.

In Figure 5, there is a screenshot from the overview of the course for the lecturer to insert related data. To create a new course or modify an existing course, first, the lecturer visits this page and needs to provide the course's general information.

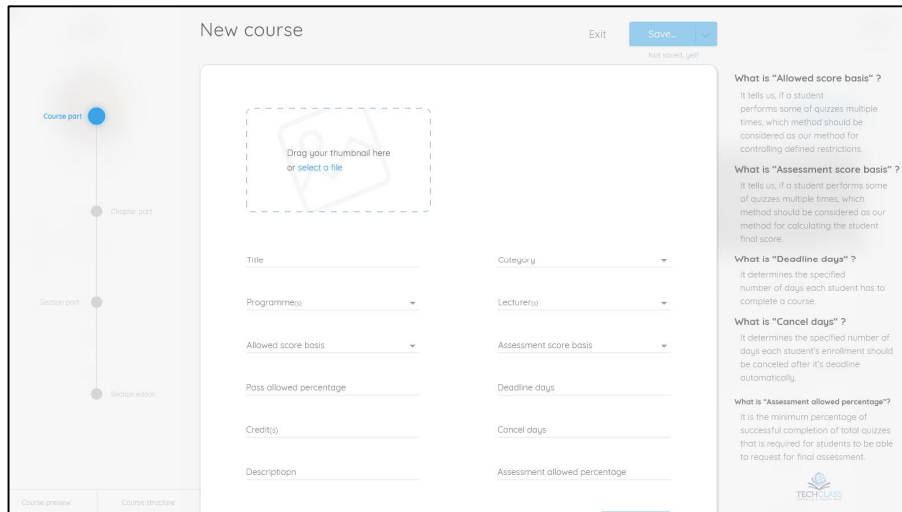


Figure 5: Defining a course in Course Designer

Figure 6 demonstrates how the lecturer can add, modify, and remove chapters, and by clicking on the chapters, the lecturer will see a similar view to manage the sections.

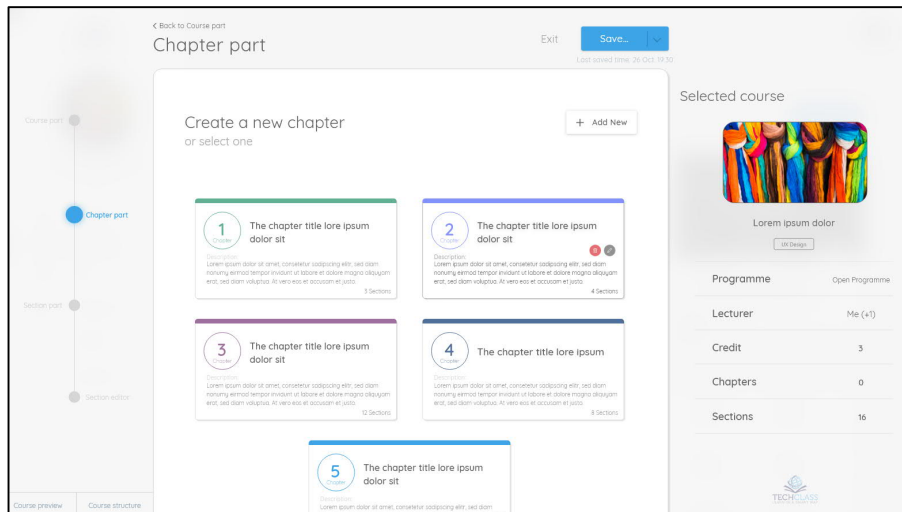


Figure 6: A sample course having a few chapters

Figure 7 explains how the lecturer can insert the content using a WYSIWYG tool called Content Editor.

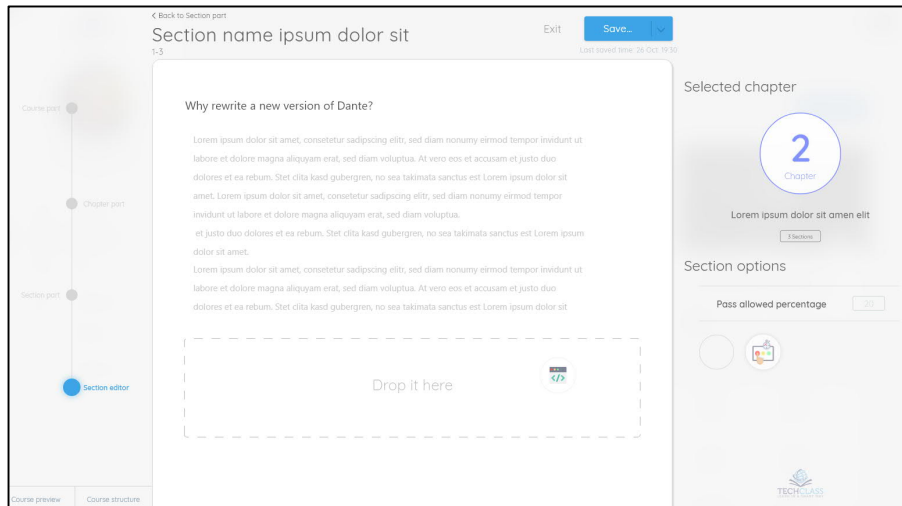


Figure 7: A section sample with content and ability to drag and drop Assignments and Quizzes

Table 2 represents the data that the lecturer can provide in the Course Designer. It is also visible from which part we ask the lecturer to provide such data.

Table 2: Data to provide from Course Designer for a course

| Data | Description | Part |
|--------------------------------|---|---------|
| Cover image | This image represents the course in many lists and the header of Course View and Course Space. | General |
| Category | Students may sort the courses based on their categories on the Courses page. | General |
| Programme | Linking courses to the programmes and when students subscribe to specific programmes can see their linked courses on the Course page. | General |
| Lecturers | Define which lecturers are the owners of the courses, and they will all have ultimate ownership privileges. | General |
| Allowed score basis | Unused property | General |
| Assessment score basis | If a student takes quizzes multiple times, this setting will define if the assessment consider first try, min, max, or the average | General |
| Pass allowed percentage | For automatic assessment, if the lecturer inserts grades for each task, the system can automatically set an enrollment as pass or fail. | General |
| Deadline days | Number of days that the students have time to work on and complete their enrollments for this course. | General |
| Credit(s) | The number of credits. | General |

| | | |
|--------------------------------------|--|----------------|
| Cancel days | After the expiration date, the system allows the student to have full access to the course and request a deadline extension. | General |
| Description | A few sentences about the course and what the student will learn. | General |
| Assessment allowed percentage | After a certain percentage, the students can request for assessment. | General |
| Chapters | List of the chapters of the course. | Chapters |
| Sections | List of the sections of each chapter. | Chapters |
| Sections content | WYSIWYG view of the section content. | Content Editor |
| Tasks | Adding two types of tasks; 1) Assignments 2) Quizzes | Content Editor |

Here is the list of issues we listed about Course Designer, which we need to fix and improve:

1. "Course part" in the navigation part is the wrong title. "General" would be a better choice. Also, no need to mention "part" for each section.
2. Uploading one single image and using that on headers and thumbnail size is not the right UI/UX approach.
3. Cancel days is not necessary, and having deadline days will have the same and more definite purpose.
4. The description field is too small since the lecturers usually provide one or two paragraphs about their courses.
5. It is impossible to drag and drop a section from one chapter to another. The lecturer should copy and paste the content into a new section and then remove the old section.
6. Allowed score basis is a new property that we need to remove.

3.2.3 Course View

In Course View, students can have a general overview of their enrolments, send a deadline extension request, cancel their enrollment, and obtain their certificate if they completed their enrollment. Figure 8 indicates a screenshot from Course View.

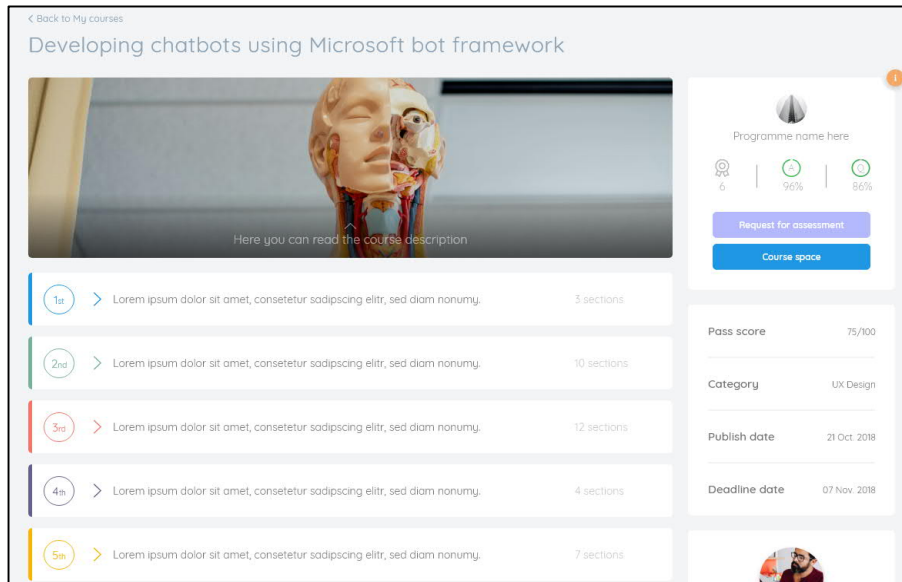


Figure 8: Course View

Here is the list of issues we listed about Course View, which we need to fix and improve:

1. If we reimagine Course Space, we need to make sure the new Course Space will also serve the purpose of Course View, so we remove this page and make the learning environment simpler.
2. We need to transfer all the necessary functionalities to Course Space.

3.2.4 Course Space

There is a significantly improved Course Space feature for the students to assist them with their tasks and following the course material throughout the learning process. TechClass aims to make the portal as smart as possible to minimize the work of the lecturers to assess the students' tasks and enable the lecturers to focus on improving their material and providing new courses. Figure 9 demonstrates a screenshot from Course Space of a sample course with various Chapters.

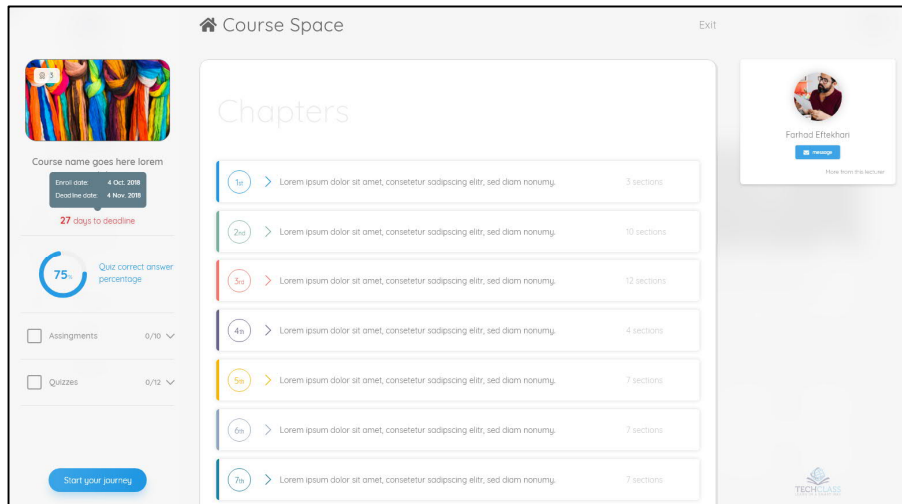


Figure 9: Course Space

Here is the list of issues we listed about Course View, which we need to fix and improve:

1. Too much space is getting wasted on the sides, and some of the indicators are using too much space, which can become reasonably smaller.
2. Content tracking is vague and not user-friendly, making it complicated for the students to browse the content.

3.2.5 Mobile view

Like other parts of the platform, Course Space and Course Designer has a suitable responsive and mobile-view status.

3.3 Analysis

3.3.1 Consistency

Before analyzing and designing and implementing the next generation of TechClass learning environments, we need to understand and tackle a significant issue. There was consistent functionality throughout the platform, except for the learning environment part.

Places such as Managing Programmes, Managing Enrollments, Managing Requests, Managing Courses, Managing Zones, and Analytics followed a particular approach, layout, and vision. To start working on the enrolled courses, users suddenly encountered a whole different environment based on the design and user experience. The reason for such a difference was that the Course Space feature was one of the first features we implemented. At the same time, we started working on the 4th generation of our learning environments. We learned many beneficial facts throughout implementing the platform, which was complicated and vague, and aiming for a perfect approach, was too optimistic. Large-scale projects such as TechClass Smart Online Learning Community require effort to get to such discoveries. Therefore, once TechClass released the new platform's MVP in February 2019, we had

the privilege to reimagine the platform's first and primary features while we were also receiving beneficial feedback from the users.

3.3.2 Business

The main reason behind having the learning environment part the way it was and not updating it sooner was that there were already many students as the platform users working on their enrollments. The company's business model was based on offering courses through online offerings of the university partners. The students register for the courses. The company invites the registered users into the platform, and they start and complete their enrollments.

TechClass could not interrupt the service and focus on reimplementing the functionalities. Therefore, we needed to find suitable timing for such replacements.

Understanding such a key factor is vital to keep the business going forward and keep having growth and progress.

3.3.3 MVP mindset

The minimum viable product is considered one of the most relevant lean startup techniques. [25] While we were implementing the 4th generation of TechClass learning environment and migrating from the 3rd generation, we had an MVP mindset. Having an MVP mindset requires a team to aim for the essential requirements and gradually build their solution around such an essential core. By making the goal too large and too complicated, teams may make their success chance lower and deprive themselves of learning throughout implementation by receiving constructive feedback. Such constructive feedback will allow essential changes to the direction of initial decisions toward a more suitable angel.

3.3.4 New Features

The first layout of the Course Space made it complicated to integrate with other systems or implement new features on top of it. The platform needed to provide more innovative functionalities such as features like Discussions, Learning Paths, Reports, etc., since the goal of releasing MVPs are usually introducing the first edition of products to receive beneficial insights and lead the direction of further developments toward a more suitable and business-aligned direction.

To implement these features, we needed to implement a reliable and unify structure. To achieve such a milestone, we needed to break the current layout, think out of the box, and innovatively re-create the application's required spaces and flow. However, more importantly, we needed to carry out the analysis from the beginning to ensure we will meet all the criteria and requirements we were aiming for and try our best to keep the future required changes to the minimum level.

3.4 Design

3.4.1 Flow

Nowadays, being user-friendly is a vital characteristic of modern and widespread application. Currently, the user requires to follow a precise flow to carry out a simple task. We needed to make the capabilities available and easy to access.

Consider a student logs in to the platform and is on the Dashboard page. To start learning a section of the course, here is the flow, which she needs to follow:

1. Click on the course
2. Getting redirected to the Course View page, where she can see the table of the content, then she needs to find the Course Space button on the right side to enter Course Space
3. Click on the left arrow to open the Course Space status panel, where she can find the "Continue Studying" button
4. Click on the button and get redirected to the Section content

The student needs to follow four various steps on four different pages. Such a procedure will make the student lose concentration and spend unnecessary time to find her way on the platform. The time that can get used in learning rather than getting redirected to various unnecessary places.

Therefore, we need to fix the flow issue in the new implementation.

3.4.2 Design Systems

A design system is a compilation of reusable components, supervised by clear that creates more significant elements such as web pages. [26]

At the beginning of implementing the platform, most elements were designed based on what we required on that single page without considering other platform parts. Unfortunately, it led to making the platform's progress complicated. It was nearly impossible to add new features to the platform without a tremendous amount of extra work because of a static design concept without any room for improvement and future updates.

We decided to use a design system to prevent all these stated issues that we were encountering.

Designing small components instead of web pages helped us develop a whole new dynamic design environment that could fulfill our needs. It carried so many options to create new features for the future of the platform.

Besides that, communications between the product design and development team became more efficient, all because of small components created instead of giant mock-ups that needed lots of meeting sessions and extra documentation to keep everything aligned across the production team.

Figures 10 and 11 show screenshots of some parts of the design system we implemented to create the features mock-ups.

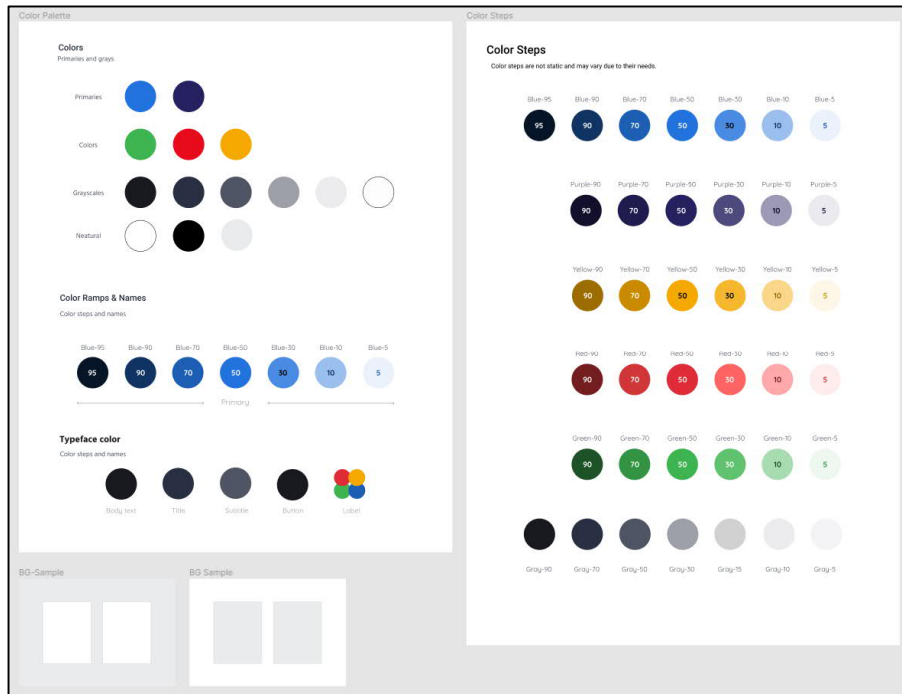


Figure 10: Color styling in the design system

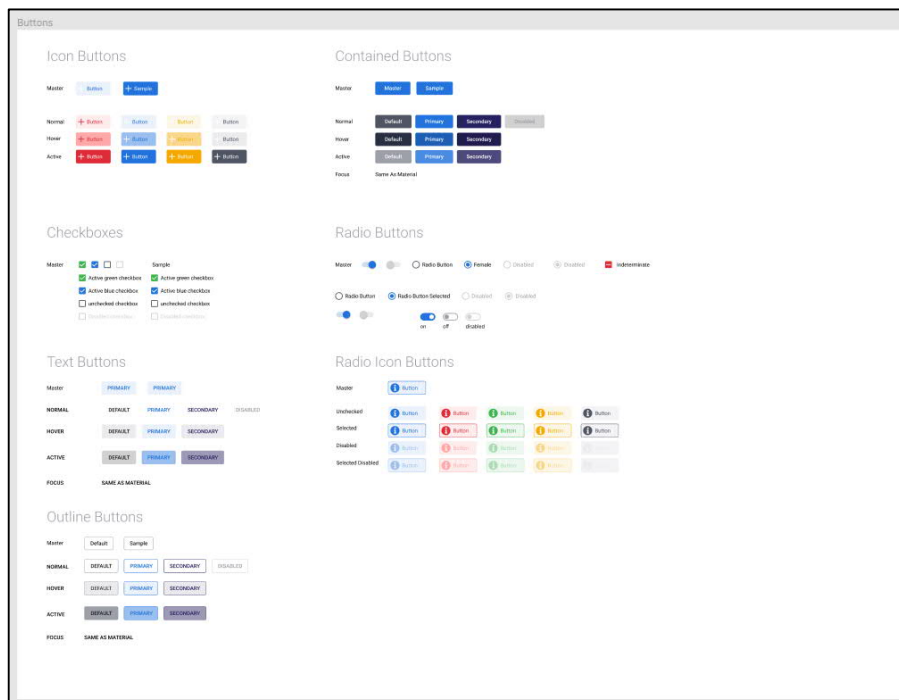


Figure 11: Design of some of the components in the design system

3.5 Development

The implementation of the platform started with ASP Web Form and Bootstrap 3, and the MVC approach. Although the first version worked adequately, there was no chance of improvement and development. Thus, in the next generation, we chose React.js and .NET core, renowned technology, and a supportive community.

Switching to a comfortable cloud structure using docker, object storage, and Redis open a significant opportunity for us and better performance and scalability.

This new approach has not ended in technical matters. We also updated the flow and process using Git flow, CI/CD pipelines, and Azure DevOps. Such approaches helped the team and product in many ways. Git is an open-source and free distributed version control system to efficiently and efficiently handle everything from small to large projects. [27]

3.5.1 API-first approach

API-first architecture is a technique in which the first order of business is to create an API that puts the target customer's needs first and then develops the product on top of it, whether it is a website, mobile app, or SaaS software. [28]

Implementing back-end was the first step of the new approach. The API project used ASP.NET Core 1 as the first stable release of the all-new Microsoft ASP.NET Core and continues to update and progress with the framework updates until now. Project follow RESTful API specification. ASP.NET Core is a cross-platform, high-performance, open-source framework for building modern, cloud-enabled, Internet-connected apps. [29]

Using RESTful API made it possible to extend the project from a web platform to a mobile, desktop, or even IoT app. A single API for serving in all platforms reduces the cost of implementing and opens a gateway to endless possibilities. The N+1 problem is a well-known issue in RESTful API to avoid that TechClass API follows JSON:API speciation with some customization.

REST is an acronym for REpresentational State Transfer. It is an architectural style for distributed hypermedia systems. [30] In the RESTful API N+1 problem, the client needs one request to get recourse, then N request to recourse child or dependency resources. For instance, the client wants to show a course, so first, it requests the course info from API, then the client must request each course lecturer to complete the needed data. This cycle could be continued. In the JSON:API specification, the client can request what data it needs from each recourse using a standard query string based on RESTful API. It is a straightforward way to solve the N+1 problem, enhance the performance (the client gets the exact data it needs), and add advanced data querying possibility.

For Authentication and Authorization API, follow the OAuth 2 protocol. The API returns a JWT token that includes user info and access scope with a signature of API. The client must send this token in the request header of each request. This approach is not limited to use in the browser. It can be used in any other future platform like mobile apps.

API requires a database to store and query data. We chose Microsoft SQL Server for this purpose. To increase query fetching performance, we used Dapper and EF core to retrieve and update the database. The query performance is an essential matter in the project. Therefore, we used query profiling tools to ensure query performance. The platform courses usually include many media; therefore, we selected Object storage to avoid storing such data in the database. Microsoft Azure provides a cloud Object storage solution with high accessibility. All the media stores in the Object storage containers and links to related rows in the database.

3.5.2 Writing testable code

We believe that developing new software is not just about accomplishing the given time frame's business requirements. Therefore, producing a maintainable, extensible, and testable codebase that meets business requirements is a primary goal in our new implementation.

On the front-end side, we selected Jest and Enzyme tools to write unit tests for most of the implemented components. Also, the Cypress framework is responsible for running all of the end-to-end tests before each deployment. Cypress is a front end testing platform of the next generation designed for the modern web. When testing modern apps, we discuss the main pain points that developers and QA engineers face. [31]

In the back-end, we use xUnit to implement and run the unit and integration tests to ensure everything works as it should. The purpose is to cover almost all the vital logic of the app.

These tools give us high confidence that our application's critical paths are always working before publishing new features into the production. The CD pipeline runs all of the tests. The new version will be released only if all the tests have been passed.

3.5.3 Choosing a modern front-end development solution

TechClass uses the ReactJS library for implementing a high-performance, sophisticated and featureful UI based on component-based and testable parts. React is a JavaScript library for building user interfaces. [32]

We also use Redux as the state management system to reduce the data complexity and manage the state shared across different components.

Besides handling the front-end side effects and moving the API call and asynchronous logic to the outside of component scopes, the redux-saga middleware is added to the project. It allows us to write asynchronous code that looks synchronous and is very easy to test.

We selected TypeScript over Flow.js for static type checking. Furthermore, such selection enabled us to discover many issues in the development phase. TypeScript is an open-source language that builds on JavaScript, one of the world's most used tools, by adding static type definitions. Types provide a way to define an object's form, provide better documentation, and allow TypeScript to verify that your code is running properly. [33]

3.5.4 Performance enhancement

One of the first decisions we made was using JSON:API, a specification to send data query from the client to the server to improve the API performance. To adjust the platform's specification, we had to implement a few customizations on top of that. The result was promising, which decreased the request count, the database hit and even requests response size.

The server responds according to the client's demands and sends the exact data requested. For instance, if the client needs the lecturers' info and their courses' name. There were two solutions here. First, the client requests the lecturer's info, then for each one requests the courses one by one ($N + 1$ problem) or makes separate endpoints for these requirements something like 'GetLecturerWithCourse'. ' which lead to $N+1$ endpoints. None of them was satisfying.

Also, in those scenarios, the server responding with all detail for requested data. The client receives full details of lecturers and their courses instead of getting the requirements, which are lecturers' info and courses' names.

In the TechClass API specification, the client identifies the shape and the relation of required data, such as 'lecturer, lecturer.courses', and then specify the required data like 'lecturer=firstName, lastName' and 'course=name'.

A cache layer was added to the project with a JavaScript service worker to decrease the requests and speed up the front-end project. A service worker performs as a middleware between the platform and the web. The cache scenario, which has been developed by the TechClass team, made the platform work more smoothly, and the user would not be needed to wait for the server responses.

SQL Query has a direct impact on API performance to monitor and improve it, the SQL query profiler tools used during development, and on the production. A query is chosen carefully with confidence that is the best practice. also, some techniques like data pagination can be useful to control the size of fetched data

Since storing media in the database is not a good practice. All TechClass platform media store in a cloud object storage. The media is behind the CDN, and it guarantees each user gets the last media from the nearest server geographic. Each media in object storage has a unique name, and It is assumed to an immutable data. Therefore, it can cache forever on the client-side.

Both front-end and back-end projects are dockerized. Docker decreased the setup time to zero and unlocked the cloud capability.

Also, Microsoft Azure's excellent monitoring tools help a lot to monitor and improve platform performance. Query monitor, Server recourse monitor, object storage hits, request's response time, errors, etc.

Using the react profiling in the front-end project helped the team decrease the project's render time significantly. Each component is controlled not to render more than it needs. Figure 12 shows a screenshot from React Profiling Tool.

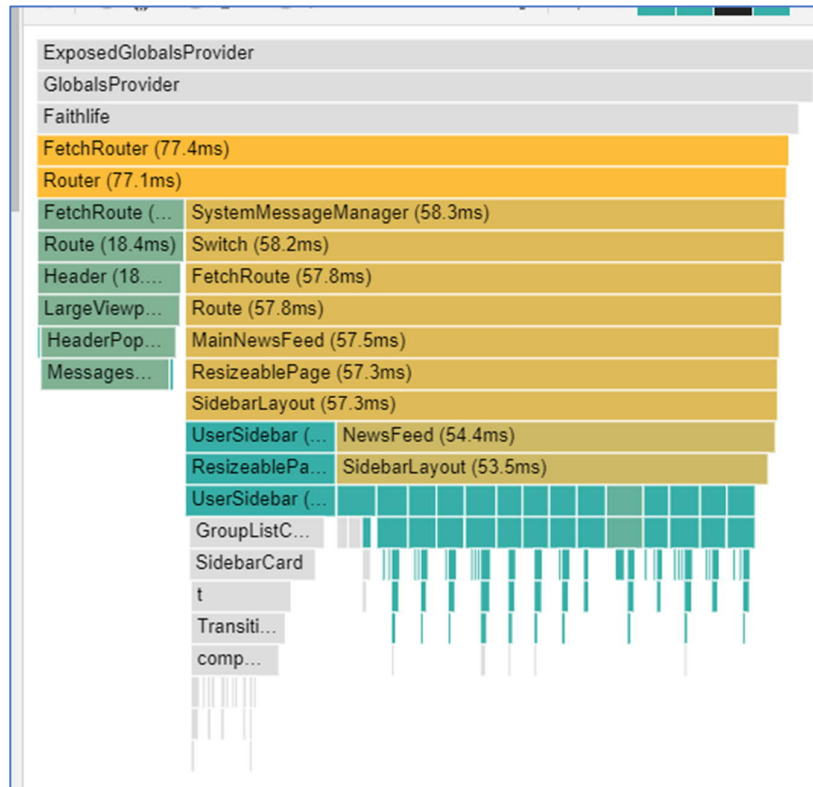


Figure 12: React Profiling Tool

3.5.5 Security

We use a JWT token to authenticate and authorize users. This token must store on the client-side, and we store it as a cookie in the client browser so no one except the TechClass site origin can read it. This way, the token protects against the CSRF attack. Because we use React, it is immune to the XSS attack by default.

To protect against SQL injection, the user inputs are never used directly in the SQL query. In the back-end project, EF Core always control and protect user input before using them at the output query, and for Dapper, A query builder packages are used to help the developer to make and reuse query and also protect the query from dangers user inputs

The TechClass API defends the server using an API call limit rate. Any client has a limited capacity to call the API endpoints in one second. The last limitation info is announced to the user by the standard header's value on each server response. Any request violates this rule will be ignored, and the server responds to it with a standard HTTP error code.

There is a more severe limitation for the Authentication endpoints. In this manner, after failing five attempts to log in, to protect users' accounts against the brute force attack, the user account will be lockdown for an hour, and any attempt to logging in even with the valid credential would be prevented. The platform handles this situation by sending an email to the users to inform the users and sending a one-time-use link to unlock the account.

3.5.6 Handling logs and errors

We use Sentry in the front-end for monitoring and logging all the client-side issues. This way, developers can debug all the errors and exceptions that occurred in the client's browsers.

In the platform back-end, we use Azure Insights, as Figure 13 demonstrates, to monitor the running applications. It automatically detects performance anomalies and provides powerful analytics tools to help us diagnose all exceptions and issues.

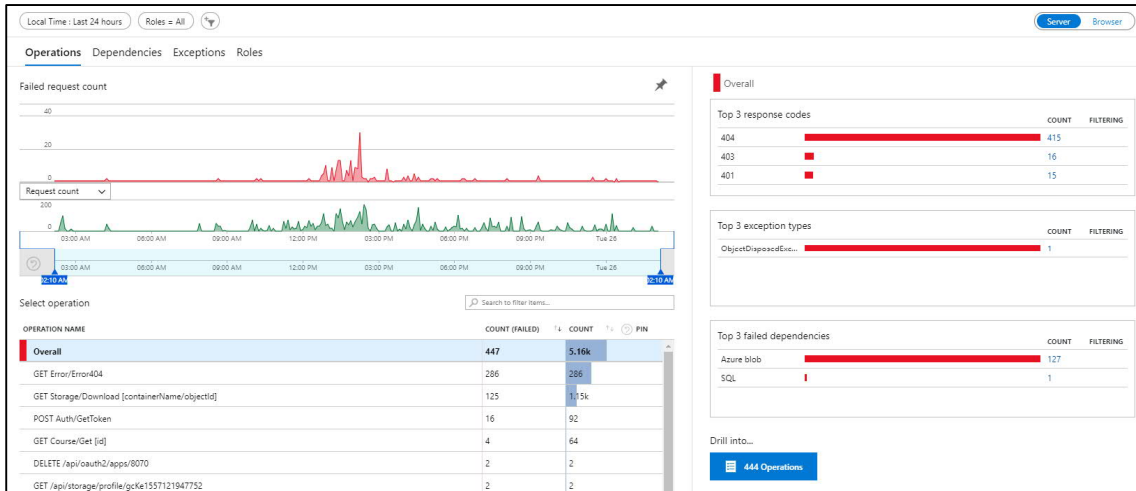


Figure 13: Azure Insights

Figure 14 illustrates a screenshot from Sentry, which we install an instance of it in our Azure infrastructures. Sentry lists the errors, which occur on the users' side. It also provides full detail regarding the number of times specific issues happened and beneficial descriptive information of the users' environments.

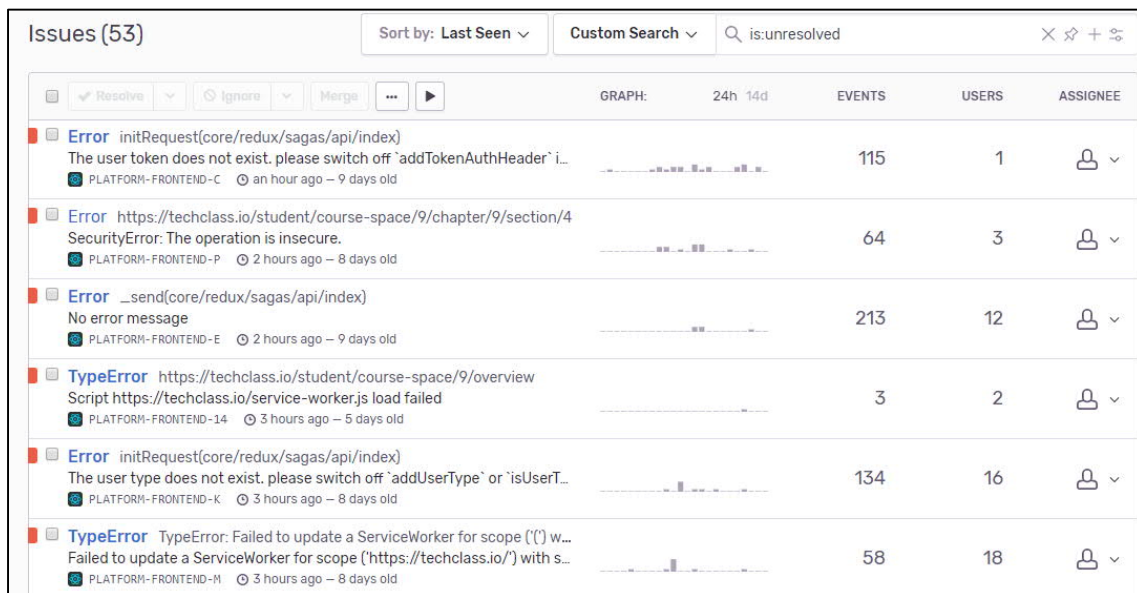


Figure 14: A sample of information provided from Sentry

3.5.7 Automate the deployment

The dev team uses Git and Azure DevOps to design robust CI/CD pipelines. The different environments such as development, quality assurance, demo, and production environment automatically build and send to the Docker registry in the Azure cloud after any changes to related branches in the git.

We use different layers of tests during the pipeline, database migrations, etc., to ensure the new release's health. Azure Docker registry will call a webhook after saving any new image. That webhook is related to the environment app server and announces that a new image is available.

Azure runs a new instance of the app bases on the new release and transforms all connections on the old one to the new one. After that, Azure will destroy the old instance.

3.6 Quality Assurance

The entire development process is prone to weaknesses and improvements in quality. It is QA's role to keep this at the front of everyone's mind.

3.6.1 Improve test coverage

We believe that quality needs to be the focus of every start. As QA is responsible for identifying any faults in the system, the QA team started testing at the initial phase when the requirement or design documents are available. To achieve this goal, we divide the testing process into two steps:

1. **Verification:** this step includes checking documents and design to ensure that the software correctly implements a specific function. At this level, the QA team begins to design a test scenario. They communicate with the team member (developers, design, and product owners) to get a clear picture of the new environment and find bugs early in the development cycle.
2. **Validation:** at the end of the development, to determine that the new environment, which we built, satisfies customer requirements or not, the QA team in this step have to update test cases according to the previous step and newly developed environment that is run in a test environment to avoid missing of test scenarios and try to cover almost 100% of the system.

3.6.2 Performance and Availability metrics

It is essential to monitor performance and availability metrics in QA. Figure 15 demonstrates a screenshot from Azure showing the health of some resources.

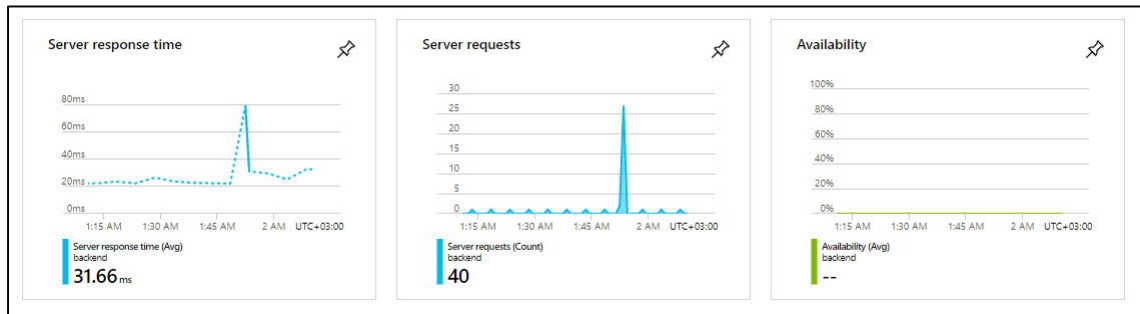


Figure 15: Resource Health in Azure

3.7 Data safety solution, GDPR, and Legal Affairs

One of the most important novelties within the EU General Data Protection Regulation is the right to data portability, both in terms of granting data subjects access rights and in terms of finding themselves at the intersection between data protection and other areas of law (competition law, intellectual property, consumer protection, etc.). [34]

As before, we consulted our legal advisor throughout the whole procedure to ensure all the implementations will be according to the GDPR and similar regulations in the EU.

User privacy is a very critical matter in the scope of our services. Follow to GDPR, access to the user data is very strict. The user password is fully hashed, and even database backup is encrypted. In the development and debugging purpose, a user data protector function converts all users' data to anonymous info before making it accessible to the developers or the QA team.

There are several backup scenarios to keep the platform and users' data safe. Azure made a database backup, and also, there is a customs procedure for periodical database backup to risk reduction of data loss. A database backup will get generated and stored in the Amazon S3 object storage at the end of each day. The Amazon S3 was chosen due to avoid a single point of failure in the system.

4 Proposed Method

4.1 Development Method

SDLC stands for the Software Development Life Cycle and is the software industry method used to analyze, develop, and test every piece of software developed. SDLC's goal is to organize the software development process's core processes to be of the highest quality in the situation. A combination of available budget, software business cases, development time limits, plus the professional level of the software architects and engineers involved in the process. [35]

Planning:

At this stage, there are significant questions that we need to find answers. Does it start with who is going to use the software application? And then, followed by, how is the software application going to be used? Finally, what information is the software going to process?

Requirement Analysis:

Analysis of requirements is the most crucial stage at SDLC. It is usually carried out by senior members of the software development team and experts from the marketing and industry. This is the crucial part of the project. The software development team's leadership must understand the essence of the software to be developed, the business case specifics, and the potential positioning of the software being developed against competitor products assuming they exist on the market.

The result of this stage is usually the SRS document, which stands for Software Requirements Specification. Identifying the development risks and the baseline of the quality assurance methodology is decided at this stage. The result of the analysis stage is to define the technical solutions that will result in a minimal risk to the project's success. [36]

Design:

The software design clearly outlines the product's architectural modules and data flow and communication diagrams within the product itself, plus specifies any third-party integration relevant to the product.

Implementation:

The complexity of this stage depends heavily on the outcome of the two preceding stages. The better the SRS and SDS, the easier the software engineers will develop the software modules they require. It is no secret that if the software's prototypes/versions are required within a short time scale, this can adversely affect the final software quality. As well as having a properly prepared SRS and SDS, quality also depends heavily on the analytical capabilities of each individual taking part in the coding process.

Testing and Integration:

Although this stage is called testing, in real life, the situation is that faults found in the software in the testing phase lead back to the development phase and then back to the testing phase in circles until the software finally reaches the necessary quality.

Maintenance:

The whole software development process can also be planned over several iterations, especially when it comes to Agile. The focus is to release a working product as soon as possible and implement different features later on.

4.2 Analyzing the Flow

As we discussed in the 2.3.1 Flow section, we realized it is hard for them to find their way to the content section to learn the material. Here are the interview session results that we hold to analyze the behavior of ten students. We considered the following measurements:

- How many seconds does it take for the students to find their way from the Dashboard to the Content page?
- Do they get lost and they ask critical questions?

These students are working with the platform for the first time; therefore, they do not have pre-knowledge how to find their way. Figure 16 illustrates the result of the survey.

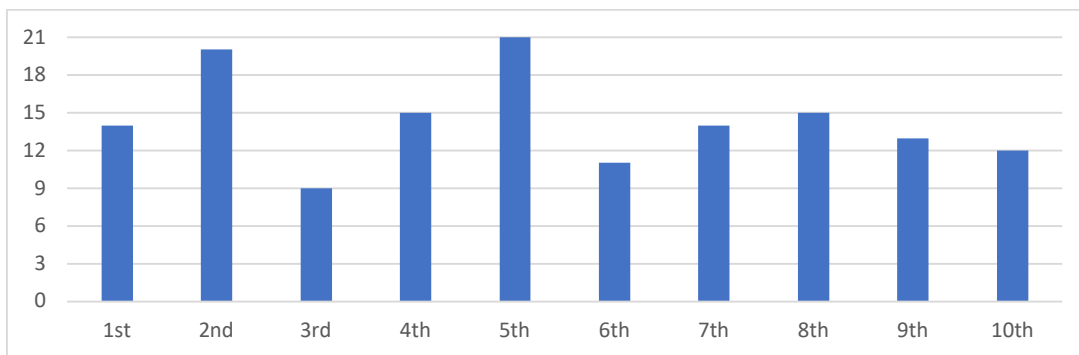


Figure 16: Each student's time (second) to find his/her way to the content part

Three out of ten students were confused, and they asked for help, all during the Course View part, which is one step before entering Course Space. One stated that he did not realize a button to direct them to the course space due to the button having no standard color. Therefore, he assumed the button is just a label.

4.3 Modern Platforms with Similar Mindset

Nowadays, there are many successful examples as online platforms that understanding and analyzing them can bring significant value to picturing a more robust and practical learning environment. Platforms that host millions of users, and there are quite many interactions happening among their audience.

4.3.1 Mixer

One of the great examples we analyzed and were fascinated by was Mixer, Microsoft's game streaming website. Figure 17 shows a screenshot from a popular channel on this platform.

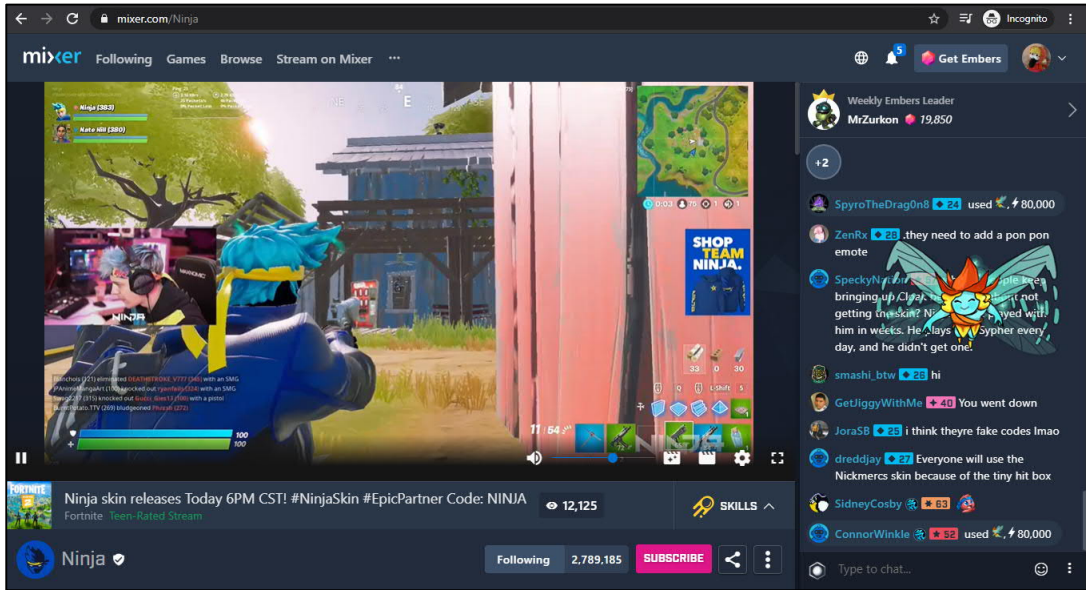


Figure 17: Mixer Gaming Platform Broadcaster Page

Taking a glance at this screenshot, you may see there are quite many elements, metrics, data, etc., all placed in one single view or page. There are a few significant parts which each has a purpose containing many elements. On the top left part, there is the logo of the platform and a few main links. On the top right part, you may see your profile avatar and indicators related to your account, and when you click on them, you can even have more information related to your account. On the center part, there is live footage of the owner game streaming channel. From time to time, informative elements pop up, and you can also control the media player. On the bottom, you may see metrics related to the user profile who owns the profile you are watching. Finally, on the right part, users communicate and chat while watching the live footage. These mentioned sections have a significant amount of details inside of them, and it is stunning to place all of such data and still experience a pleasant user experience.

Figure 18 shows the same screenshot from the Mixer platform, and in this screenshot, we labeled the various area they created to insert the related data in them.

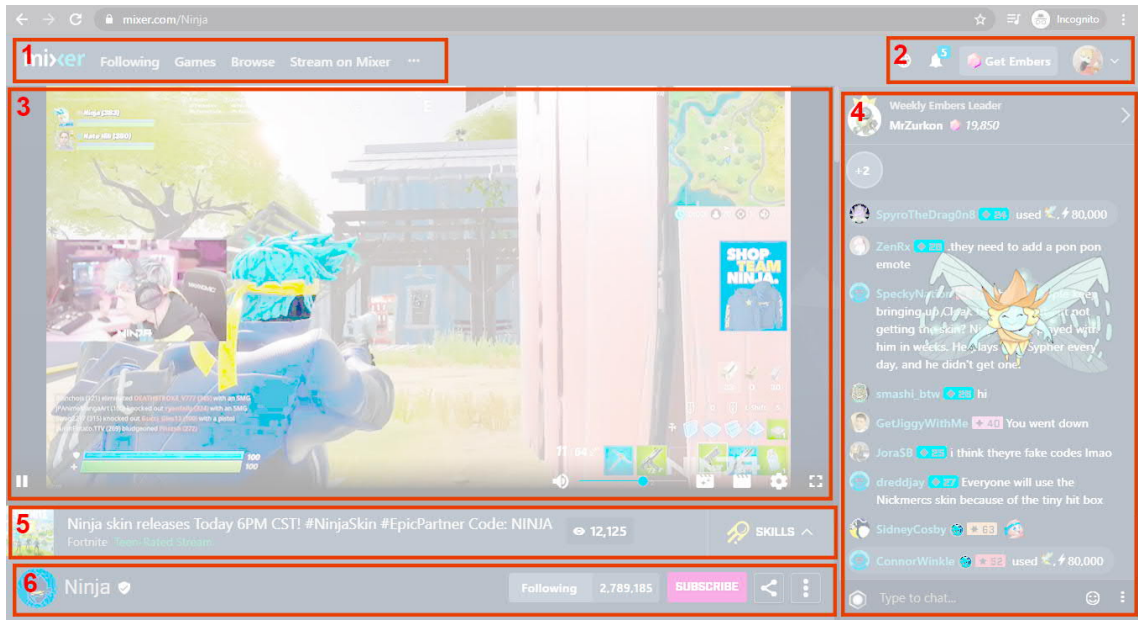


Figure 18: Mixer platform and main areas

Table 3 explains what contains such areas.

Table 3: Various areas in the UI of Mixer

| Area | Contains |
|---------------|---|
| Area 1 | <ul style="list-style-type: none"> 1.1 - Mixer logo (act as a link to the home page as well) 1.2 - platform main navigation menu |
| Area 2 | <ul style="list-style-type: none"> 2.1 User avatar (access to user settings, logout link) 2.2 Notification icon (list of notifications) 2.3 Globalization |
| Area 3 | <ul style="list-style-type: none"> 3.1 Broadcasting a channel 3.2 Control buttons |
| Area 4 | <ul style="list-style-type: none"> 4.1 Conversation history of the live users who are watching the broadcast 4.2 Text box for the user to submit a message 4.3 Place to show users tokens, which donate to the broadcaster |
| Area 5 | <ul style="list-style-type: none"> 5.1 Image of the game that the broadcaster is playing 5.1 Broadcast title that the broadcaster selected 5.2 Number of online users who are watching 5.3 Skills menu |

| | |
|----------------------|---|
| Area 6 | 6.1 Broadcaster avatar and username (verified icon if exists) 6.2 Number of total followers 6.3 subscribe/unsubscribe button for the visitor 6.4 Share button 6.5 Related more menu |
| Area 7 | 7.1 Broadcaster full profile page |
| (Not visible) | |

It is fascinating to have such an appealing design and then contain such a load of various data.

4.4 Data Models

Index B represents the data model of the learning environment.

4.5 Layout and Vision

At first, we needed to reimagine our learning environment to include innovative ideas and create new possibilities. Second, we needed to combine such innovation with what we already provide and then aim for an outcome, which fulfills both goals.

4.5.1 Wireframe

Wireframing is a way to build a service at the structural level of a website. A wireframe is widely used to set out content and features on a website that takes into account user expectations and usability testing. Until graphic design and content are introduced, wireframes are used early in the development process to create the basic structure. [37]

Figure 19 represents the first wireframe that we created to deliver the design aligned with the vision.

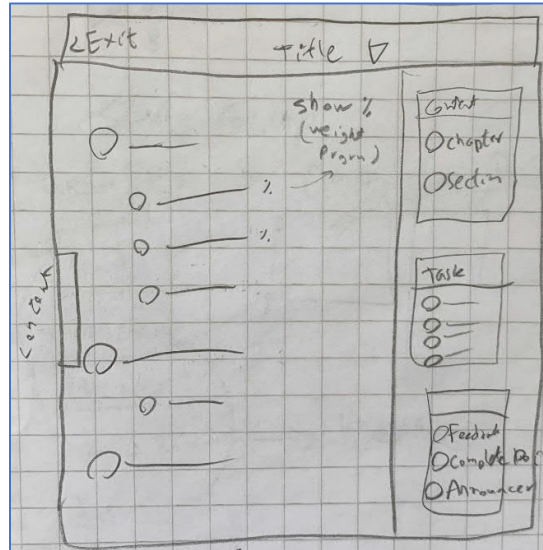


Figure 19: The first sketched wireframe

4.6 Obtaining Feedback

Obtaining feedback is essential to ensure that we understand the users' requirements, behaviors, and expectations. Thanks to receiving the Tempo funding from Business Finland, TechClass allocated more resources to receive feedback from various user-groups of many potential markets.

Timing of the feedback:

It could be more beneficial if the feedback's timing were before we finalize the new learning environment. If we could have the before and after data, a comparison of the result could help evaluate our results. Although, most of the aspects we improved, modified, or introduced were straightforward and clear that the direction is accurate.

Feedback Approach:

The learning environment survey is going to be an essential part of the feedback form. While designing the feedback form, the main concern was to receive a beneficial analysis of how the users are using the platform and what requirements we are missing from the current version of the application.

Feedback Questionnaire:

Index A presents the questionnaire that we used to obtain feedback from the various user types. The questionnaire targets three user types, student, lecturer, and staff. Section 5.11 provides the result and the analysis of the questionnaire.

5 Result

We finally implemented our all-new Learning Environment according to the feedback and analysis we had. In this part, I will explain the outcome and the reasons behind the decisions we made throughout the implementations.

5.1 Singular Vision

There are two major features, which are having a very similar origin and basis. The features are:

1. Course Space: This is where our students learn the content.
2. Course Designer: This is where the lecturers insert their content.

I conclude there should be a similar layout for these features and make the flow a centralized solution. I decided these two major features should have a similar overview because we want to give the content provider experience as a lecturer similar to the experience of a student who uses Course Space to learn the material. Such direction will enable the lecturer to ensure that he/she is aiming for a more proper structure.

5.1.1 Mock-up

Figure 20 illustrates a mock-up of the Singular Vision concept.

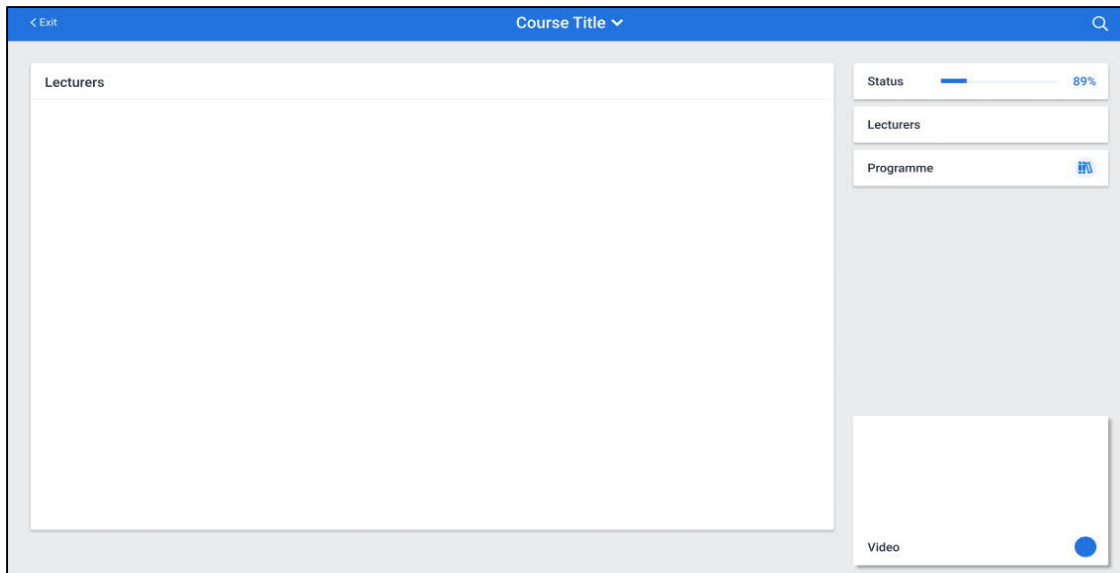


Figure 20: Singular Vision in Desktop view

Figure 21 emphasizes the main area in Singular Vision.

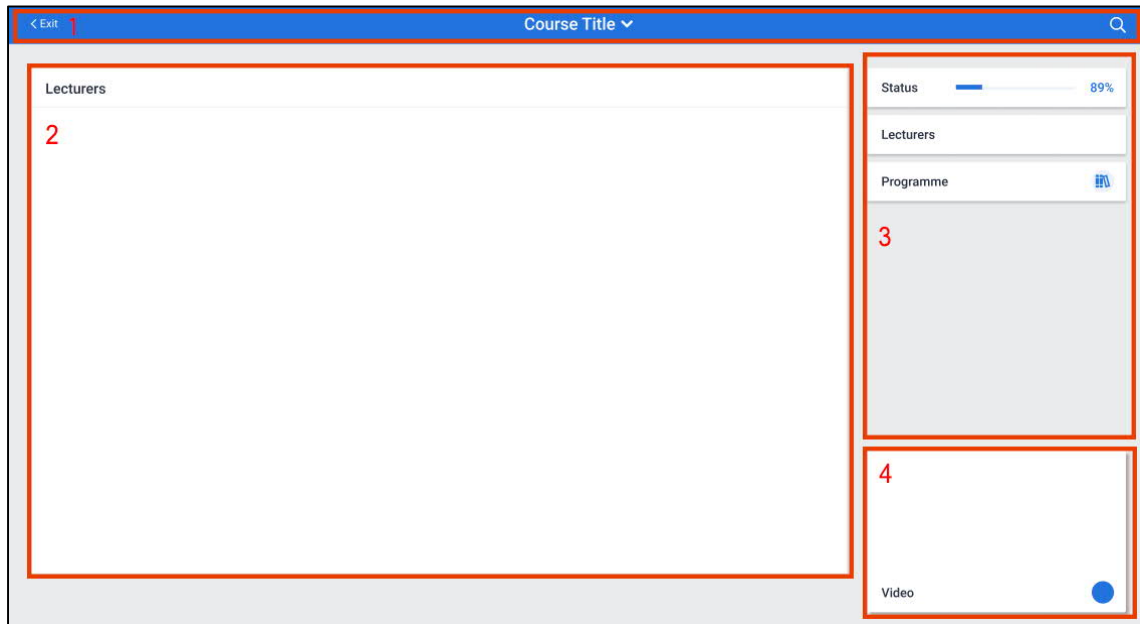


Figure 21: Singular Vision and its areas

Table 4 shows the type of data in such areas.

Table 4: Area and the data they represent

| Area | Contains |
|---------------|--|
| Area 1 | 1.1 Course title and switch drop-down 1.2 Exit button 1.3 Search button |
| Area 2 | 2.1 Center area to show items in a more spacious area |
| Area 3 | 3.1 Boxes for major functionalities and features |
| Area 4 | 4.1 Minimized version of the tutorial video with hiding and showing capability |

There are few exciting details about the layout of Singular Vision.

First, 1.1 Course title and switch drop-down enable lecturers and students to swiftly go from one course to another without the need to go back to the Dashboard or the courses page and select another course. Figure 22 demonstrates a screenshot of an open menu.

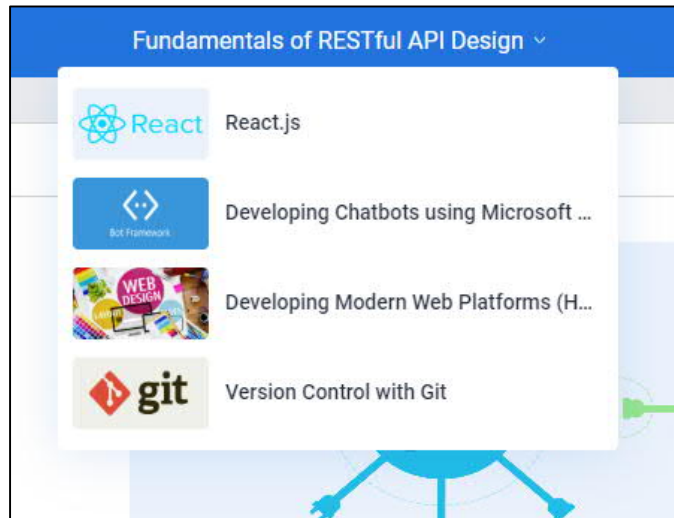


Figure 22: The top drop-down menu to navigate between courses

Area 2 specifically allocated to show the content has one purpose; allow the users to focus on the most critical affairs. Figure 23 shows a screenshot of the content of one course.

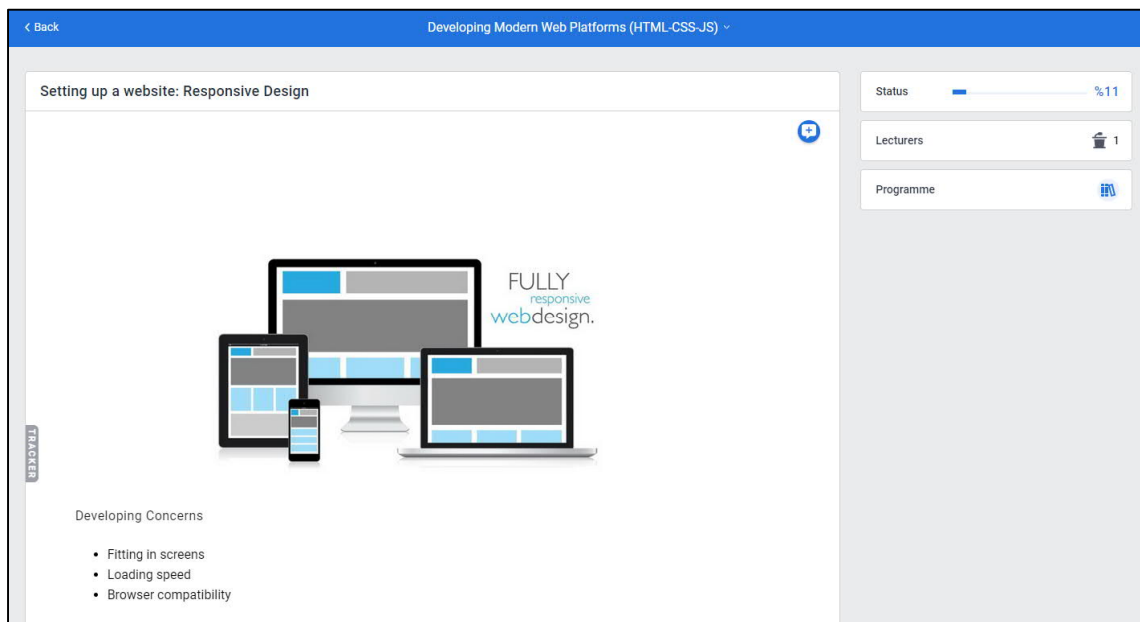


Figure 23: A view of a course in Course Space

Furthermore, Figure 24 shows a screenshot of a sample assignment in the same area.

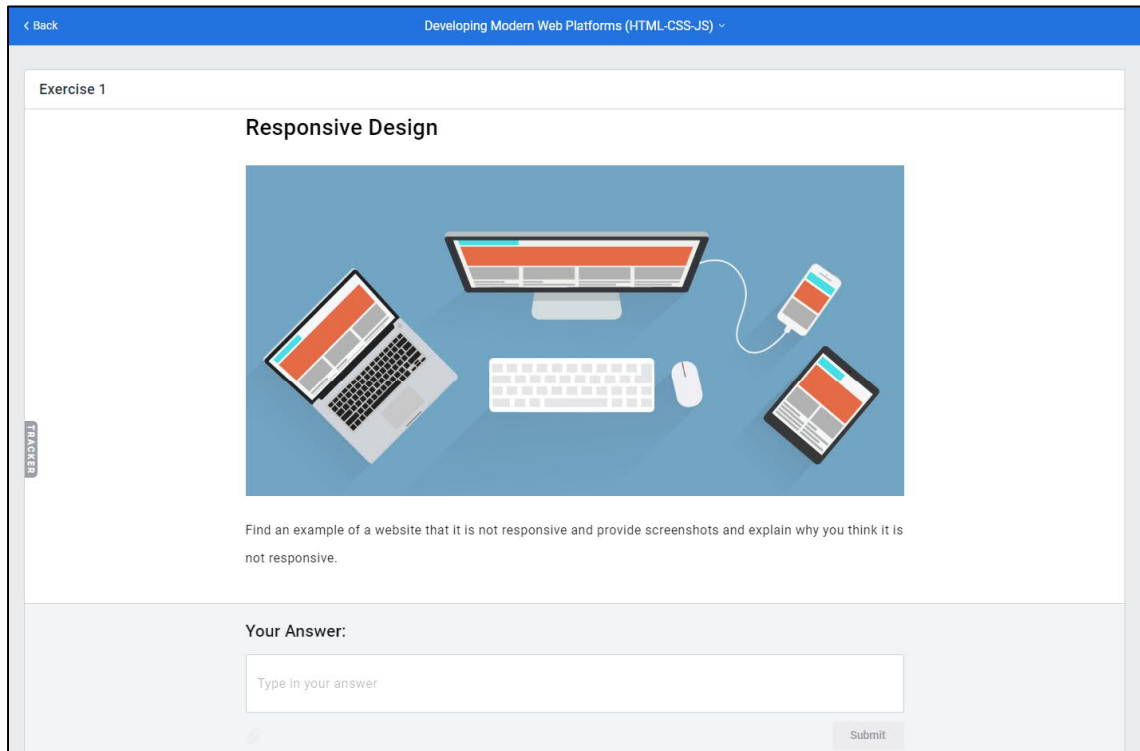


Figure 24: Assignment view in Course Space

5.1.2 Final Design

Figure 25 demonstrates the final design of Course Designer, where on the right side, it is clear the metrics and functionalities allow the lecturer to manage the course and his/her students' affairs.

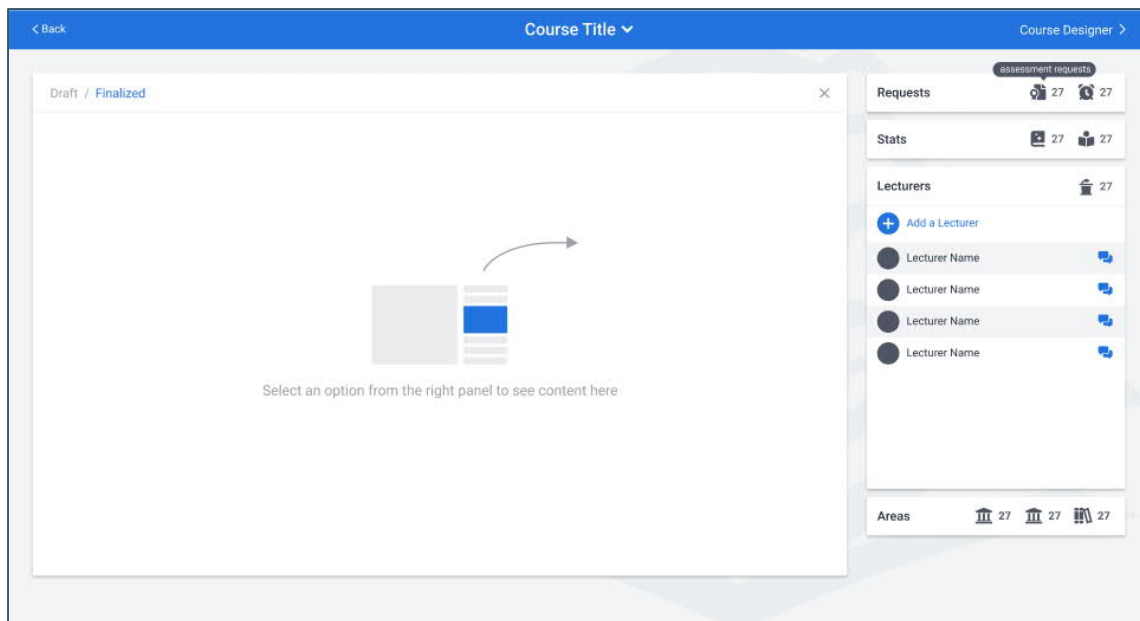


Figure 25: The final design

5.1.3 Mobile view

As a student, while working with Course Space, the platform offers a similar user experience both in web and mobile views. In the previous version, we had an ultimate mobile-view situation, and in the new version, the goal was to keep the main benefits while adding new values and possibilities.

Figure 26 demonstrates how initiative and user-friendly it is to use Course Space in the mobile view.

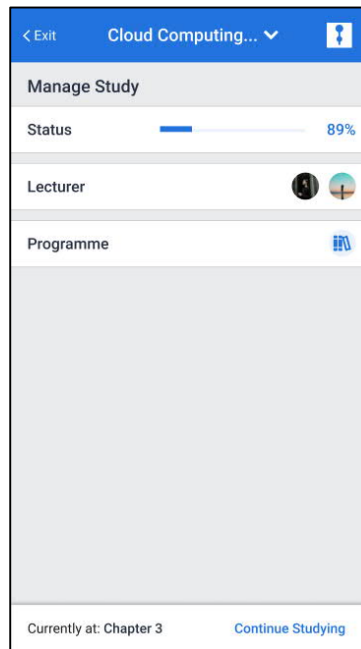


Figure 26: Singular Vision in Mobile view

There are a few major parts in this finalized layout; The top part, which makes it easy to navigate through different courses, and also using the Exit button, the user can get back to the rest of the platform. The right part includes various boxes, each for a significant purpose, and by clicking on them, the user will receive an expanded view. The center part, in which the user can view the content and do the tasks.

Unlike Course Space, in Course Designer, we do not allow the lecturers to modify the course content, as it is visible in Figure 27.

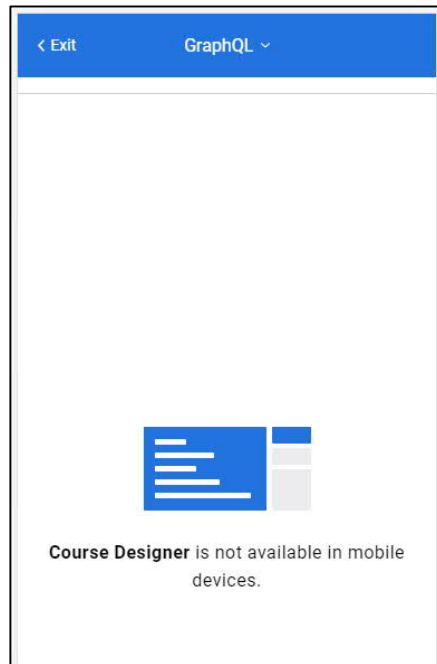


Figure 27: Course Designer in mobile view

The reason is that providing advanced functionalities that we implemented in the content editor cannot be possible. Also, we do not expect the lecturers to modify their course content on their mobile phones where on the go, they can use their laptops for such purpose and enjoy a more spacious environment.

Although we do not make it possible to use Course Designer, still the lecturers can use Course Management to manage the basic and primary aspects of their courses, as Figure 28 illustrates.

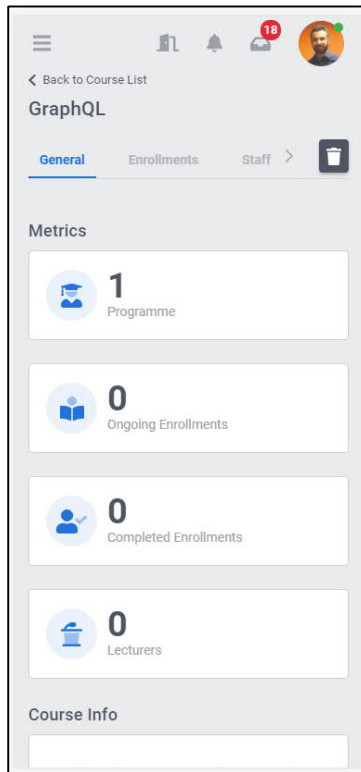


Figure 28: Mobile view of Course Management

5.2 Fixing the direction

The finalized layout significantly improved the development direction. From now, all the features may quickly get inserted and having secure and user-friendly integrations between various features.

In Figure 29, there is a red rectangle to demonstrate the space reserved for future functionalities, integrations, and any related interaction for the current and future needs in the Singular Vision concept.

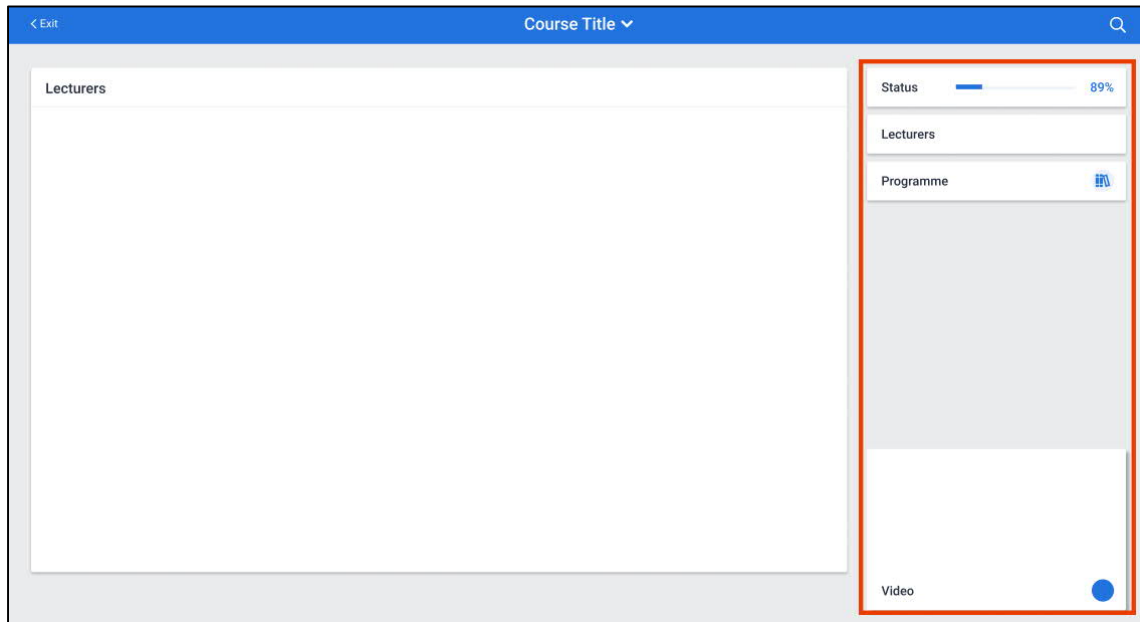


Figure 29: Reserved space in the Singular Vision concept

5.3 Design and Design System

5.3.1 The characteristics and strength of our design system

After reviewing available design systems on the internet and listing our needs, it became more evident that we could easily use an open-source, well-documented design system. A design system that we could transform into a great design echo system and get our desired result from it.

The strength of our design system is that it does not depend on external elements. Although we can consider it as a distribution of material design, it has its own rules.

5.3.2 How we improved UI and UX of the new learning environment

For many years, Learning Experience Designers did not have to concern themselves with the user interface. However, nowadays, most designers agree on the connection between UI and UX and how important it is to master both of them simultaneously.

At TechClass, we consider User Experience an inevitable part of our production, and we are proud of providing a realistic and straightforward e-learning environment. Most eLearning contents appeared within a standard 'click-next' player, with the course menu navigation aid at the left or right side. Some of them allow the user the choice of showing the course menu.

5.3.3 Comparison between the old and the new content editor

Another beneficial improvement was enabling students to use the powerful content editor. Now, students may use this feature to work on their assignments, projects, essay, and other tasks.

The Content Editor is one of the essential features of the TechClass online learning platform. It allows lecturers and student to write professional documents using an advanced, but easy to use editor.

In the previous version of The Content Editor, we delivered a well-known editing area to our users to modify and write their documents in it. It worked well, although it did not get the result that we wanted. Therefore, we immediately started to analyze the basics of the current version of this feature and wrote issues and difficulties that it was countering. Also, in this period, Google Analytics helped us a lot to get a better perspective of our users. Figure 30 shows the previous edition of the Content Editor.

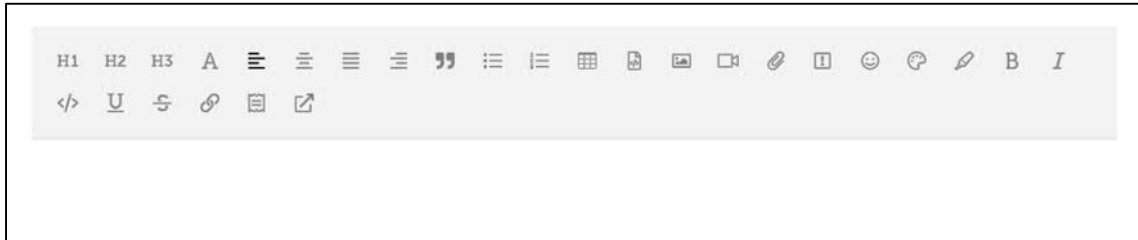


Figure 30: The previous edition of Content Editor

As shown in the above image, we used to have a fixed position toolbar that needed a mysterious place of the screen so that the user could use it to edit and style content. It was complicated, tedious, and not practical when pages of a document became more than two or three. Because the user had to scroll to the top to see the toolbar, and it was not convenient as it had to be. Figure 31 demonstrates the updated version of the Content Editor.

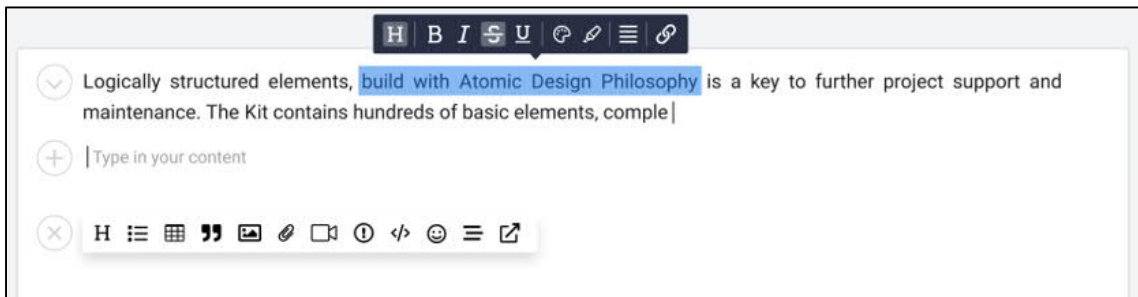


Figure 31: New edition of Content Editor

In the new version of Content Editor, our primary focus was to make writing an enjoyable task in TechClass. We replaced the old, complicated toolbar with two small mini toolbars—one for adding new stuff to the content and changing the written paragraphs' styles.

These two mini toolbars do not reserve a space. It means they are in your hands whenever you need them. For example, when you want to start a new line, a plus sign would appear and asks you what sort of content you want to enter this line. It can be a table, paragraph, heading, etc.

The other toolbar (dark bar) that has been mentioned earlier helps you to style your content. When you select some part of a sentence or a single word, it would pop up and hands

you some useful options to choose from. For instance, you can select a word and makes it Bold, Italic, or Underlined, etc.

5.4 Technical Improvement

In this section, we discuss the technical improvements in the new learning environment.

Using TypeScript and drop Flow:

In the last version, most of our front-end codebase and all the old features are reimplemented from scratch with TypeScript and react hooks.

Improving the coding pattern:

We try our best to break the front-end logic into the smaller, testable, and cleaner components and move all API calls and other side effects to the saga layers.

Writing more unit tests:

In some of the old components, many anti-patterns made writing standard tests impossible. We learned much from these issues and implemented a better version that follows the best practices and added more unit tests to the project.

Implementing a custom cache strategy:

I have worked on custom cache strategies in the front-end to cache a lot of required data in the browsers and prevent unnecessary requests.

Reducing the platform load time:

To provide a better UX experience, we used the lazy load concept, and the bundle chunked into smaller pieces. Such an approach makes the platform loads faster in the first loading phase.

Using hooks and new React features:

In React 16.8, an all-new Hook API has been released. An elegant and straightforward solution to share the component logics and manage the internal state. React Hook was a new chapter in React development and began a new thinking method in JavaScript and even other programming languages.

The development team decided to refactor all the components while changing the component design to matching the new design system with the React Hooks. This decision did by replacing small parts of the project. These changes eventually cause the whole system to be replaced. The code quality has been grown more than 50% through the refactoring, and thanks to the modern pattern and Hooks, duplication code and logics are removed entirely. During this process, more than 10000 hooks were developed and used.

Implement a new Editor with Slate 0.50:

Slate is a completely customizable framework for building rich text editors. Slate enables building rich, intuitive editors similar to those in Medium, Dropbox Paper, or Google Docs. [38] In the previous version, we used Slate 0.47 and implemented a massive amount of code and logic to scratch up all the capabilities.

Design system implementation:

Implementing the design system reusable components took a lot of time and effort.

We created most of the components and items needed to build our features using them; This makes our features codebase smaller than ever because all the duplicated and similar components are replaced with the standard one from the design system.

All new components were designed and developed with the mobile-first view idea to support more devices with different screen sizes and optimized to work smoother in these devices.

5.5 Quality Assurance Validation

The approach I adopted to achieve better quality in the new learning environment, as it is mentioned, consists of two general phases: verification and validation.

In the verification step, when the analysis and design documents are finalized, the QA team starts to create a test scenario according to these documents and check that the analytics and design documents are consistent or to find a bug in the early stage.

In the validation step, I update and create new test cases according to new features to find defects and report to the developers to cover almost 100% of the system.

In the process of QA, I perform various types of tests to increase test coverage. Some test types that are done in the UAT environment:

Smoke Testing:

Smoke testing is done to make sure that the build received from the development team is stable and covers critical functionalities. Smoke testing (also referred to as "Build Verification Testing") is a type of software testing consisting of a non-exhaustive series of tests to ensure that the most critical functions work. The outcome of this testing is used to assess if a construct is stable enough to begin further testing. It can also be used to determine whether or not to announce or reverse a production release. [39]

I picked the necessary test cases from already written test cases that cover the application's primary functionality to conduct smoke testing. In general, we picked a minimal number of test cases that will not take more than half an hour to execute.

This approach can minimize test effort, save time, and improve the application; moreover, the correction of errors becomes easier. Also, smoke testing can be done manually or automatically.

Functional Testing:

Each feature is tested by feeding its input and then verify the output to ensures that the requirements are adequately satisfied by the new environment learning. Functional testing is a form of black-box testing, in which functional specifications/requirements are evaluated against each part of the device. [40]

Regression Testing:

Regression testing identifies no defect in the system due to changes made over other parts of the system. It also makes sure old defects will not appear from the addition of new software modules over time. [41]

Responsive Testing:

According to statistics shown in google analytics, in the last three months, 45 percent of users use Apple iPhone, and five percent of them use iPad; these devices respectively come in first and second place of most usable devices. Also, about the browser, most users use Chrome. Therefore, these devices and technology are considered as a target of our responsive testing. It is done by a real device or website that is the online responsive checker.

Integration Testing:

It is a testing level to check whether all software modules, especially modules in the new learning environment, coded by different developers, are integrated logically and communicate with each other correctly. The goal of integration testing is to identify defects in the interaction between integrated units. [42]

Automation Testing:

It is essential to identify suitable candidate among test cases of a new learning environment to develop automation test scripts and prepare them before running a test; there are some criteria we consider to select suitable test cases and automate them at first:

1. Tests that take a lot of effort and time when applying manual testing
2. Repetitive tests that run for multiple builds
3. Smoke testing

Also, we complement automated tests with manual tests that are better run by humans, for example, when we need visual validation like features in the content editor.

We choose Cypress to write faster, easier, and more reliable tests to run end-to-end testing automatically among different tools for automation. It is often compared to Selenium; however, Cypress is both fundamentally and architecturally different. Cypress also is not constrained by the same restrictions as Selenium. The other advantages of Cypress are:

- JavaScript testing framework
- Easy control of response bodies, status, and headers
- Built on top of Mocha and Chai
- Helps to find locators

Figure 32 shows the overview that tests are executed in real-time on our test environment.

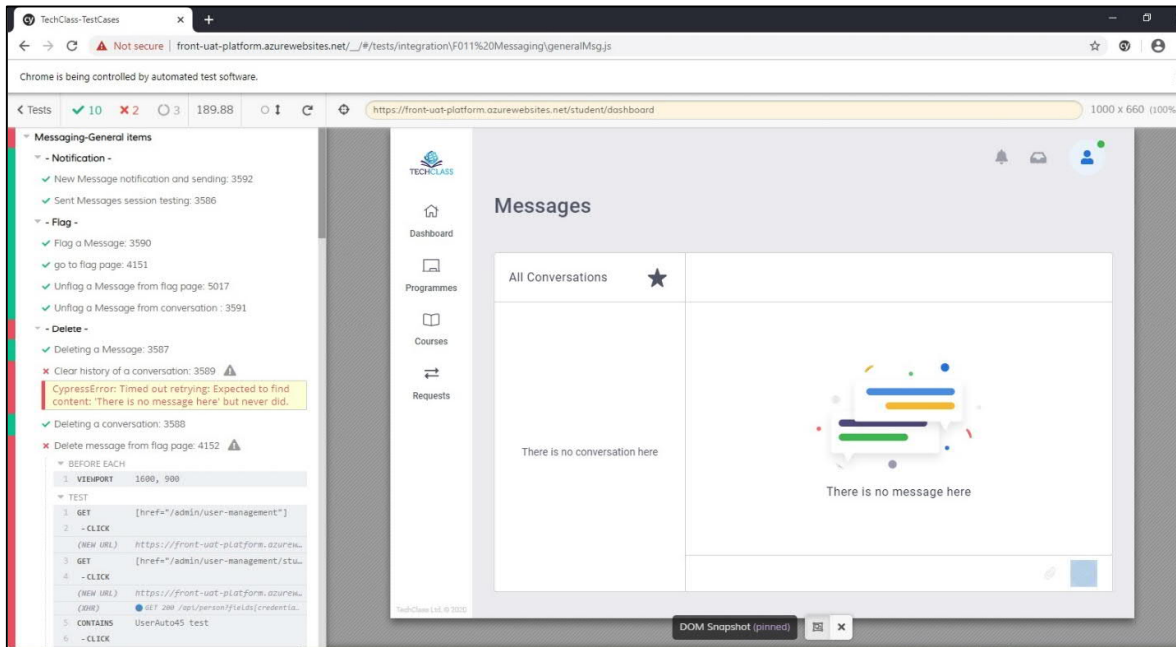


Figure 32: A screenshot from Cypress

Developing automation testing takes time and effort; therefore, at first, allocating time and resources for automation might come across as a problem. However, automated tests can be written once and run several times; it obviates the need for spending much time on manual tests, which must run each time. In a nutshell, the more we automate test cases, the more time and cost we save to perform a tricky test and improve test coverage. For example, in our project, a test suite (Messaging Feature) that contains 33 test cases, we automated 30 test cases that are run in 180 seconds (3min) by Cypress. If we want to run these test cases manually, this could take at least 15 minutes. Accordingly, if we extend that to the whole project, we perceive how beneficial automation is to apply.

Front-end Unit Testing:

As it consists of checking isolated sections of the code, as units, Unit Testing is different. Typically, these units take the form of methods, properties, the behavior of UI elements, etc. It is the easiest to incorporate and the cheapest. [43]

Figure 33 shows a screenshot from the coverage report of the Badge component.

| All files src/components/Badge | | | | | | | | | |
|---|------------|----------|-----------|-------|------|-----|--------|-------|--|
| 91.67% Statements 11/12 83.33% Branches 15/18 100% Functions 1/1 91.67% Lines 11/12 | | | | | | | | | |
| Press n or j to go to the next uncovered block, b, p or k for the previous block. | | | | | | | | | |
| File | Statements | Branches | Functions | Lines | | | | | |
| index.tsx | 91.67% | 11/12 | 83.33% | 15/18 | 100% | 1/1 | 91.67% | 11/12 | |
| types.ts | 0% | 0/0 | 0% | 0/0 | 0% | 0/0 | 0% | 0/0 | |

Figure 33: A screenshot from the coverage report of the Badge component

It means that the test only considers that the child object is always undefined or null, and it just executes the other part of the code: “<span className= ...”. This report shows that the test covers all functions, including the inner functions of the Badge component.

Back-end Unit Testing:

The back-end project requires a comprehensive refactor to apply trending approaches better and improve the code's performance and readability. Therefore, since we knew we have to go through such a phase, and many parts of the code will change, the back-end unit testing status is not suitable.

5.6 Integrations

The new layout and structure of the Course Space and Course Designer made it significantly easier to implement various integrations. We have an efficient structure on the back-end and lots of usable space on the front-end to apply integrations.

5.6.1 Microsoft Teams

Microsoft Teams is a continuous forum for chat-based collaboration, complete with document sharing, online meetings, and many other extremely useful business communications features. [44] We are targeting Microsoft Teams as the first primary tool to integrate into our system. Microsoft Teams is a powerful tool for organizations to use for communication purposes between teams. Although we have practical built-in communication features in our platform, connecting our platform to Microsoft Teams will open an ocean of possibilities. We can implement three apps:

1. TechClass Learn: Learning Environment (Student)
2. TechClass Teach: Teaching Environment (Lecturer-Tutor)
3. TechClass Manage: Managing Environment (Staff-Admin)

Table 5 shows what kind of integrations we can implement for the various features and MS Teams mobile Apps.

Table 5: Various integrations with MS Teams

| Feature | Functionality | TechClass Learn | TechClass Teach | TechClass Manage |
|----------------------|-----------------------|-----------------|-----------------|------------------|
| Notifications | Receive Notifications | v1 | v1 | v1 |
| Dashboard | /stats | v1 | v1 | v1 |
| Messages | Receive Messages | X | X | X |
| | Reply | X | X | X |
| Request | Take Action | | X | X |
| Zone | Join | | | X |

| | | | | |
|--------------------------|---------------------------------------|---|---|---|
| Course Space | Submit assignments | | | |
| | Take quiz | X | | |
| | Get Grade and get result | X | | |
| Certificates | Get Certificate | X | | |
| | Someone got certificate | | X | X |
| Analytics | Ask specific analytics | | X | X |
| | Receive monthly analytics | | X | X |
| Course Management | Notify changes | | X | |
| | Enrollment key got generated | | X | |
| Discussions | Received a discussion to moderate | | X | |
| | Received a reply to a discussion post | X | X | |
| Support System | Someone created a ticket | | X | X |
| | Receiving a reply | X | X | X |
| License Manager | Ask the type of license | X | X | X |

5.7 GDPR and Legal Affairs

To be more aligned with the GDPR, since we were not using or benefitting from storing the users' phone numbers and addresses, we decided to drop such data and do not ask the users anymore.

5.8 Features to Implement

Fixing the direction in the learning environment section and preparing the platform enabled the team to get more prepared to introduce advanced features and functionalities. A few of the features that we already analyzed and finalized the design and mock-ups will be discussed in this section.

5.8.1 Discussions

While a student is learning the material, they can start a discussion and post their message there if there were any questions. Later the lecturer can reply to the post. When adding a discussion, the user can select what the type of discussion is. Users can select between a few choices; error, confusion, extra material are choices as the discussion type.

Discussions can have two management choices that lecturer can choose:

- Open: As soon as a student post a discussion, it will be open for all to view and reply

- Moderate: First, a lecturer of the course needs to approve the discussion, then the students can view them.

Table 6 also explains the privacy types which lecturers and students can select.

Table 6: Discussion posts various privacy modes

| Type | Selectable by the student | Selectable by the lecturer |
|-----------|---------------------------|----------------------------|
| Public | Yes | Yes |
| Anonymous | Yes | Yes |
| Hidden | No | Yes |

Figure 34 demonstrates the design of the Discussion feature from the student point of view in Course Space. In this design, two major parts are essential. The first part is on the right side of the content section, where there are discussion icons. Students can add a new discussion using the plus button, and for viewing the existing ones, by clicking on them, a panel from the right will be visible, loading the data of the discussion post and its replies.

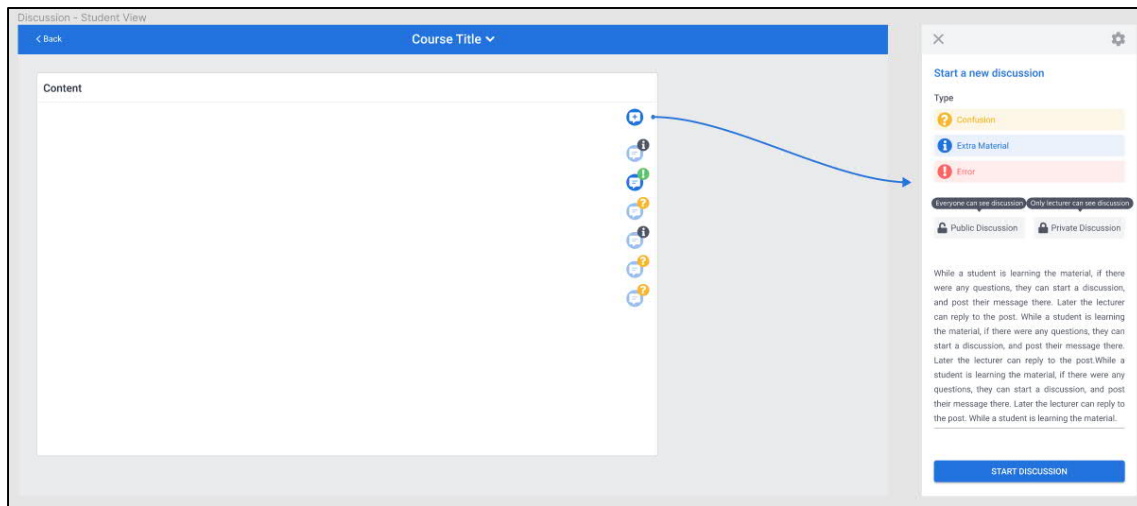


Figure 34: Discussion view in Course Space

Figure 35 also suggests a view where the lecturer can manage the discussions.

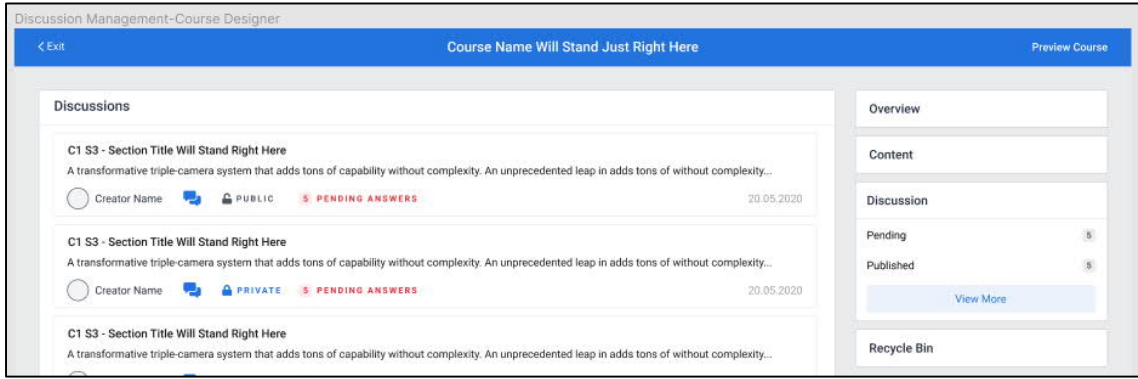


Figure 35: Discussion management section for the lecturers

There will be a section in the Course Management for the lecturers to decide the type of discussion. Figure 36 shows the part related to the Discussion feature management in Course Management.

Enrollment Type

Choose one of the enrollment Types

Public
 Request
 Access Code

Course Discussion Type

Choose one of the discussion Types

Open
 Moderate

Figure 36: Discussion in Course Management

5.8.2 Shared Sections

We needed to allow the lecturers to save an individual section and it for various courses. For instance, the intro and finishing sections of the courses may have identical sections where the lecturer introduces herself and the teaching style. When the lecturer modifies such sections, the changes can easily be affected in several courses by having a master section.

Figure 37 suggests a design to manage Shared Sections in the course design table's content part.

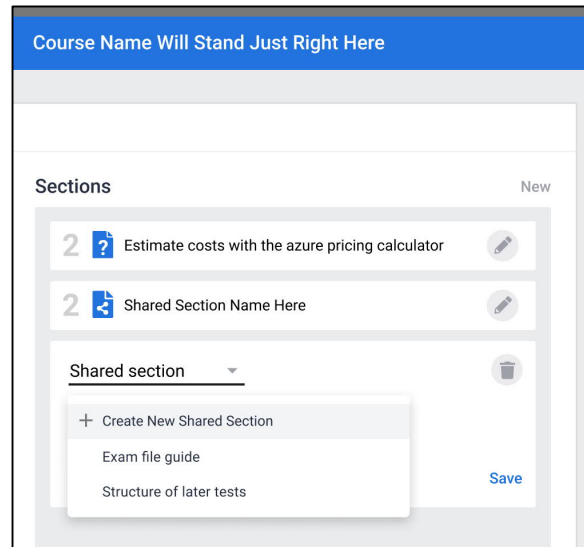


Figure 37: Shared Sections feature in Course Designer

5.8.3 Learning Paths

We need to let the lecturer select difficulty for tasks and sections, so when students are studying the material, select how deep they like to learn. According to that, the platform shows its targeted content.

In Course Designer, we need to have a section for the lecturer to setup difficulty steps and order them. Here are the items we need to include in Learning Paths:

- Order
- Name
- Credit
- Optional message

The lecturer also needs to define if the students can change their selected level while studying and still have not completed their enrollment.

Figure 38 demonstrates how students can select an option as a Learning Path when enrolling in a course.

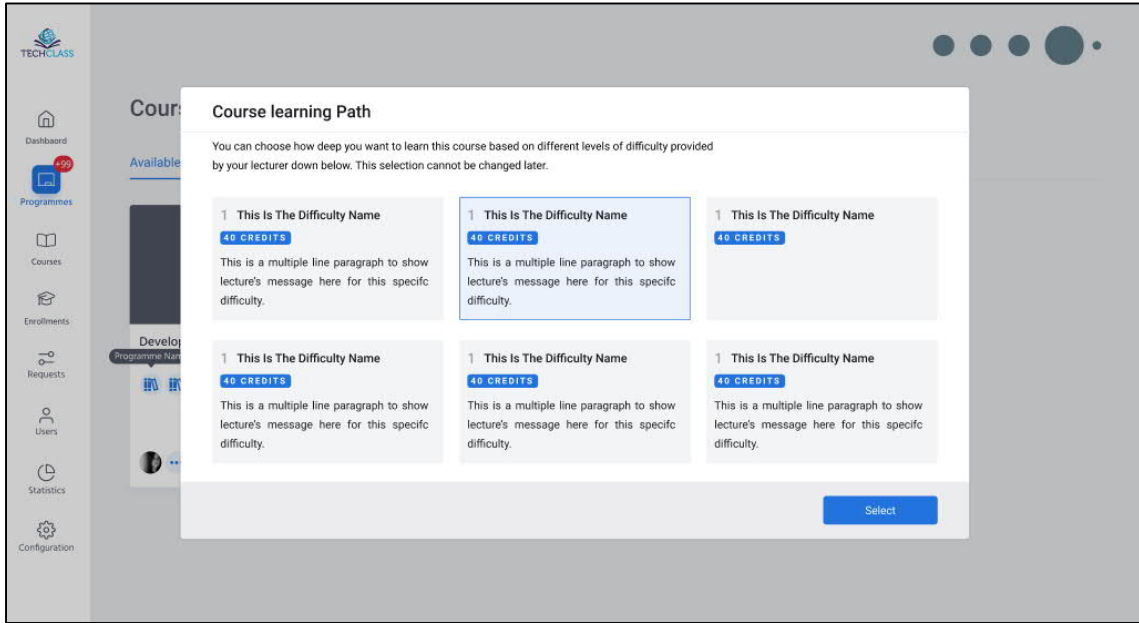


Figure 38: Selecting an option as a Learning Path

In Course Space, Status box, the students need to view the level they selected, and if the lecturer is allowed, they can change the level. Figure 39 suggests such a place for managing Learning Paths from the students.

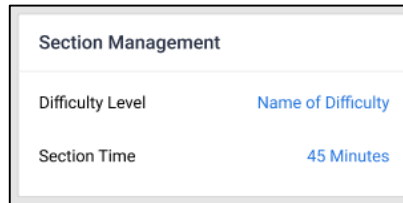


Figure 39: Managing Learning Paths by the students

5.9 Timeline

Implementation of the new learning environment consisted of 5 major tasks. Table 7 represents the timeline of the project for various tasks.

Table 7. Timeline of the project

| # | Task | 9.19 | 10.19 | 11.19 | 12.19 | 1.20 | 2.20 | 3.20 | 4.20 |
|---|----------------|-------|-------|-------|-------|------|------|------|------|
| 1 | Planning | _____ | | | | | | | |
| 2 | Analysis | | _____ | | | | | | |
| 3 | Design | | | _____ | | | | | |
| 4 | Implementation | | | _____ | | | | | |

5 Testing and Integration

6 Maintenance

Since the project's purpose was clear, the planning phase took a short time, but for most of the project, Analysis, Design, and Implementation continued. Furthermore, once there was an output to test, we started the Testing and Integration phase, and finally, it took about two months to maintain and finalize the project.

5.10 Questionnaire Results

As I described in 4.6 Obtaining feedback section, we sent a questionnaire, and in this section, I represent the result we obtained. The participants are coming from Nigeria (the first 15 participants) and Finland (the rest).

Question 1: How easy is the learning environment to use?

This question aims to investigate the usability of the product. Figure 40 represents the result.

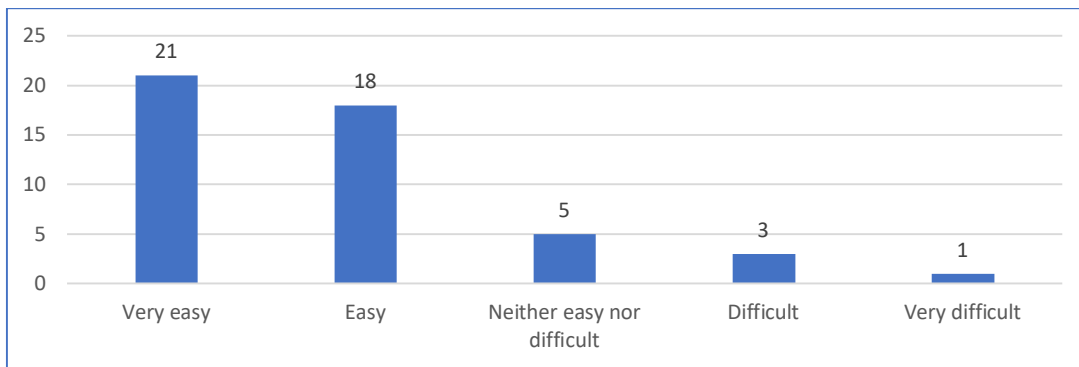


Figure 40: Question 1 result

As you may see, the result is satisfactory. If we take a look at the result of each participant in Figure 41, we may see the slow speed of internet connection had a significant impact on the feedback of the participants from Nigeria.

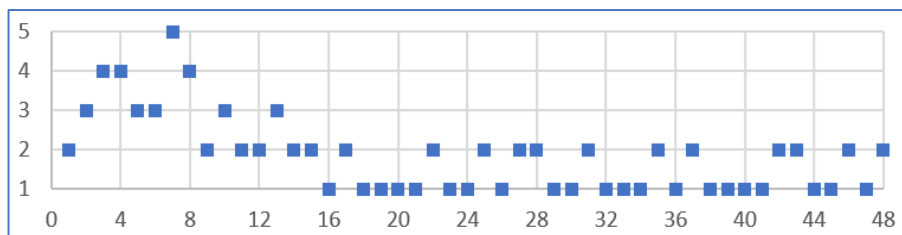


Figure 41: Question 1 result from each participant

The first ten people are from Nigeria, where the internet connection is much slower compared to Finland. We implemented the new platform considering the high-speed connection

of the internet in Finland. Therefore, in our future implementations, we learned to make the website more optimized.

Question 2: How satisfied are you with the look and feel of the learning environment?

This question gives a practical insight into the UI/UX of the product. Figure 42 demonstrates the result.

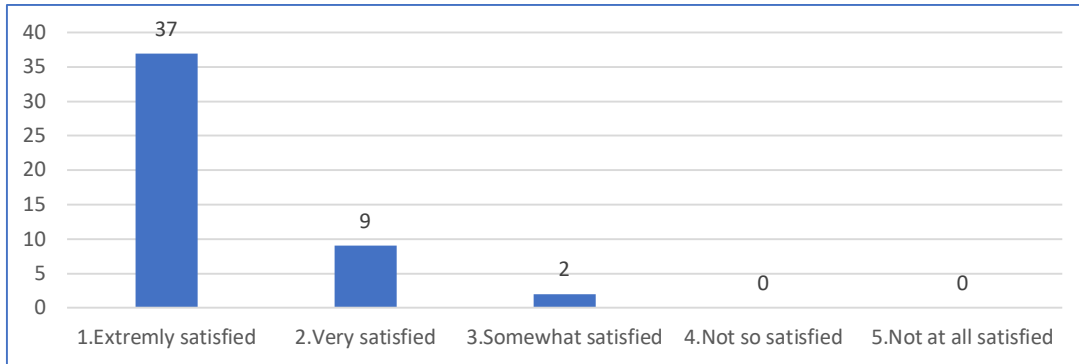


Figure 42: Question 2 result

The result of this questions is very satisfactory. Figure 43 also demonstrates the results from each participant.

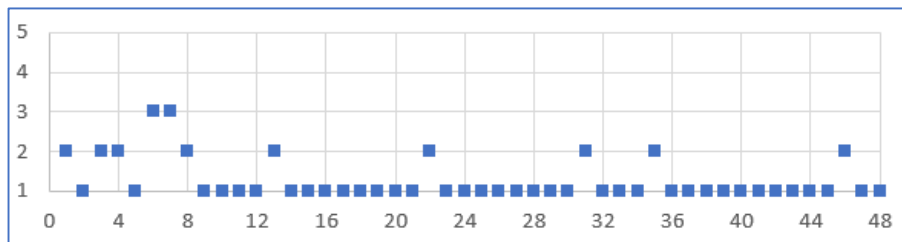


Figure 43: Question 2 result from each participant

Question 3: How satisfied are you with the performance, loading, and responsiveness of the learning environment?

This question focuses on the performance, responsiveness, and related affairs of the product. Figure 44 shows the result.

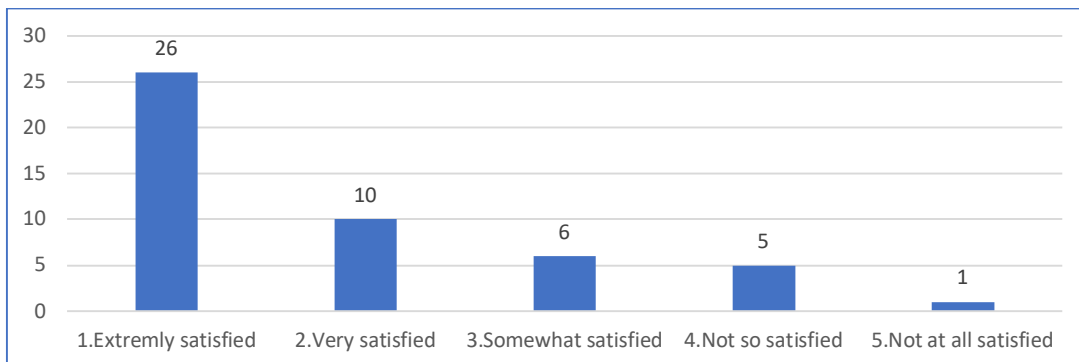


Figure 44: Question 3 result

In general, the result is satisfactory. We are also facing the same issue that we had with previous questions that the slow internet connection of some of the participants had a negative impact. Figure 45 demonstrates the results of each participant.

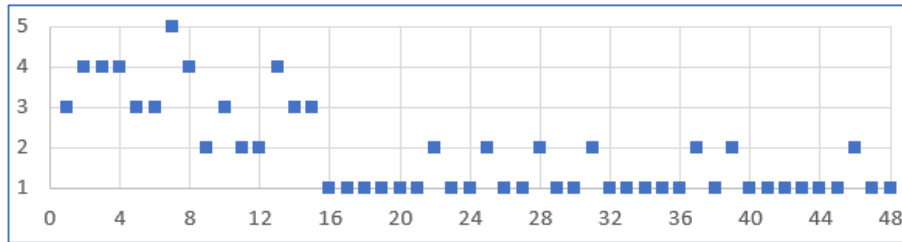


Figure 45: Question 3 result

Question 4: How well does the learning environment meet your needs?

This question delivers a general overview of the product. Figure 46 delivers the result.

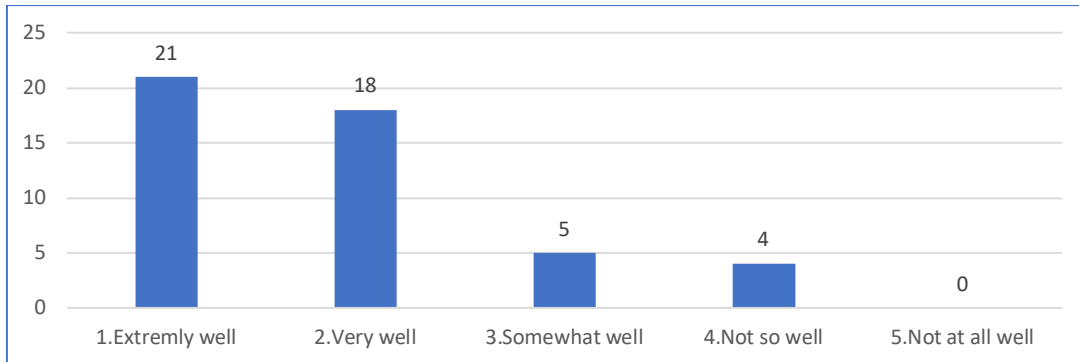


Figure 46: Question 4 result

In general, the result is satisfactory. We are also facing the same issue that we had with previous questions that the slow internet connection of some of the participants had a negative impact. Figure 47 demonstrates the results of each participant.

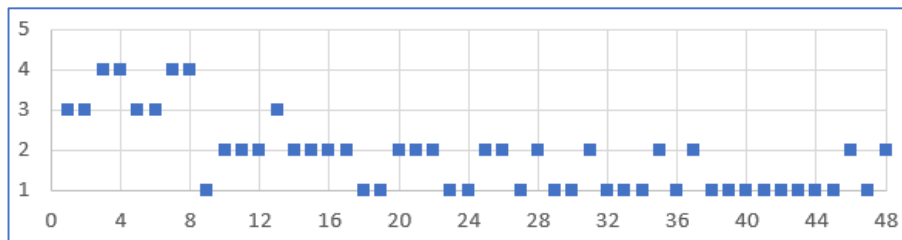


Figure 47: Question 4 result from each participant

Question 5: Which of the following words would you use to describe the learning environment? Select all that apply.

This question delivered various aspects that the users understand from the characteristics of the product. Figure 48 presents the result.

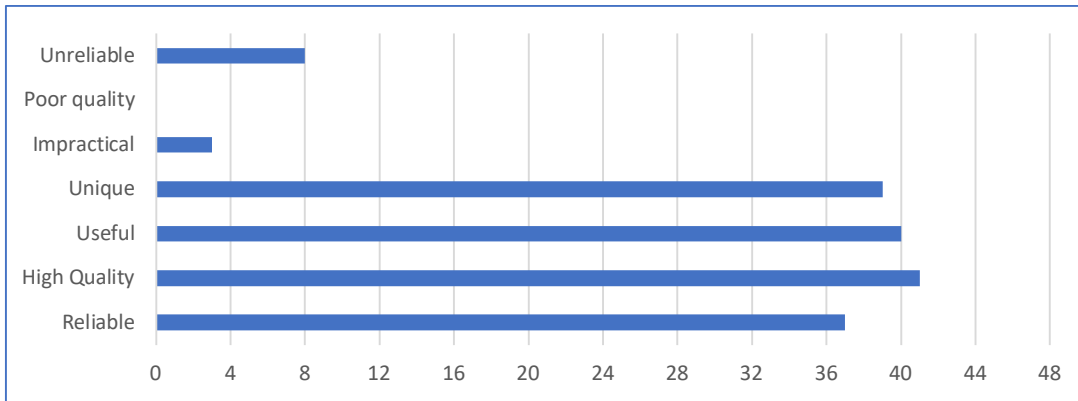


Figure 48: Question 5 result

Question 6: How likely is it that you could recommend the learning environment to a friend or colleague?

This question shows user feedback directly, and it is also a popular way of measuring customer satisfaction. If a user does not recommend a product to their friend, they may feel disappointed in some aspects.

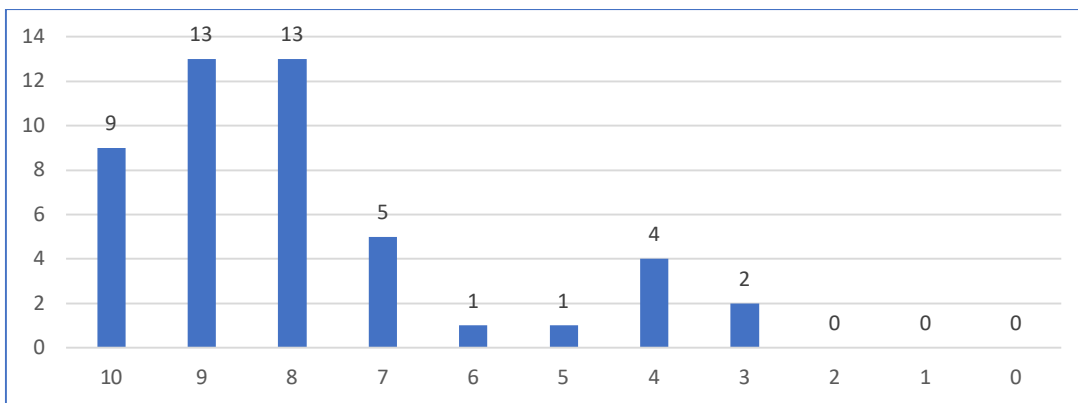


Figure 49: Question 6 result

In general, the result is satisfactory. We are also facing the same issue that we had with previous questions that the slow internet connection of some of the participants had a negative impact. Figure 50 demonstrates the results of each participant.

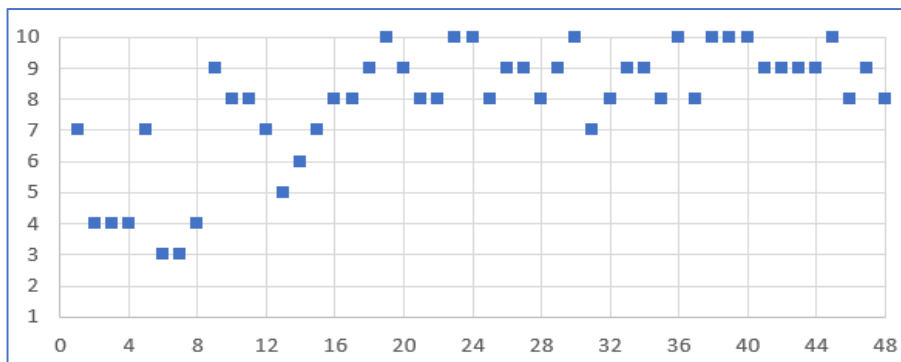


Figure 50: Question 4 result from each participant

6 DISCUSSIONS

This part will discuss a few relevant and related subjects according to the research goals we introduced in the introduction section.

6.1 Limitations

In this part, I will discuss the limitations of the previous learning environment before introducing the Singular Vision. Finally, I will discuss the current limitations that we have and need to improve in upcoming releases gradually.

6.1.1 Previous learning environment edition limitations

There are some major concerns regarding the online learning environment. One of which is that students' challenges lie more in the problems of interacting on online platforms efficiently and appropriately. Students should aspire to determine the value of peer input and discussions, along with the idea of working together via shared online learning environments. [45]

Considering the previous edition of TechClass Smart Online Learning Community learning environment, here were the critical and noticeable limitations and problems:

- Having many unnecessary and vague properties in Course Designer for the lecturer to define, such as cancel days.
- A few bad UI/UX approaches for both Course Space and Course Designer, such as mismatch of the course image's size as the course's header.
- Challenging and confusing chapter and section definition part. It is impossible to drag and drop sections into other chapters; therefore, the lecturer had to remove and entirely enter the content again in a new section.
- Course View was an unnecessary step for the students before they get into Course Space and learn the course material.
- Inefficiently using the space in both Course Space and Course Designer led the main area to be small and hard to use.
- Content tracking and browsing were noticeably hard for students.

6.1.2 Current learning environment limitations

Soon we will learn in more detail about the limitations that the current system will cause. We will be looking for such limitations since trying to tackle and find innovative solutions for such limitations is a key to improve the learning experience.

We currently have some ideas about some of the limitations because there will be no perfection in real-life projects and aiming for a complete solution is a constant cycle. Following the MVP approach, the service should have an update to release the updates, and the providers learn more about what they need to improve.

Here are some of the limitations of the current system:

- Placing of some of the properties overlap in Course Manager and Course Space

- No top-level section multimedia possibility to represent as the section's tutorial video
- Content editor can become more user friendly similar to gitbook
- Using hotjar tool will provide insight to use for better UX
- A better CI/CD plan since releases are time and resource consuming

6.2 New capabilities and upcoming features

One of the significant reasons we implemented our Singular Vision concept for the learning environments was to make it significantly more comfortable for the upcoming features to get introduced and support our Learning Environment. There are many features already lined up for implementation to make learning more supportive and pleasant.

Here are some of the upcoming features:

6.2.1 Smart Sandboxes

To enable the students to learn easier with fewer obstacles and enabling the platform to carry out the lecturers' manual work as much as possible, we will introduce Smart Sandboxes features.

Type of languages/technologies:

- Need a compiler: It is more costly and challenging to set up the environment for these languages due to their use of more resources. Examples are C#, SQL, and Python.
- Does not need a compiler: These languages do not need any communication with the back-end. Therefore, they can efficiently run on the browser of the student. In this way, it would be less costly to have them. Examples are HTML, CSS, JavaScript, and TypeScript.

Here are the various methods which we can use to launch the environments:

- Method 1 – Dockerize: We can dockerize the languages, and there are already excellent sources that we can use and modify. Each time a user wants to have this capability, we can initiate and run it. We can also isolate them, which gives us a safe environment to offer to the users. For this, we need to create an App Service and handle it from there. Then we can have docker containers for each technology.
- Method 2 – Mock: We can create a more straightforward mock server by defining some commands and acting as a CLR but not a CLR. For instance, we can use this for the Basics of Git.

Collaboration:

It would also be possible to let the users (students and lecturers) collaborate. We can even go as far as showing the cursor and live updating.

There is a good possibility that we may be able to use the Visual Studio Live Share feature.

IDE:

Visual Studio Code would be a great IDE that is open source and easily integrated into our Course Space.

Also, Microsoft has its IDE Core named Mono Editor, which is also an exciting choice, and we can empower it as well and possibly brand it as TechClass IDE.

Design:

We can use a sandbox in many ways on our platform. One practical idea would be to have an extensive editor (80%) width of the screen, alongside a sidebar for lecturer instructions and having a forced test on student implementation at the end of the section.

6.2.2 Feedback System

Receiving feedback from the students is essential, and nowadays, the learning environment should become adaptive. The lecturers need to modify their content according to the feedback they receive from the students. Course Space can easily add feedback system integration on different occasions.

Here is the list of few occasions on which the students may submit their feedback:

- The expectation when they begin an enrollment
- Final feedback when they complete an enrollment
- Exist feedback if they are canceling their enrollments
- Random occasion feedback

6.2.3 Assignments Improvements

The collaboration will get more comfortable, not only for the multiple lecturers to work on a single course but also to define study groups and work on assignments.

Once a student completes an enrollment or some tasks, when the lecturer set a grade, the system can show a report to the student how well she did then other students.

After submitting an assignment, using pattern recognition implementations, the student can receive a percentage to indicate how authentic her submission was.

6.2.4 Blended Courses

Few essential features should get integrated into the platform to allow a vital ability to offer blended courses. Features such as Calendar, section view management according to dates, attendance checker, etc. are essential to provide a classroom experience.

6.2.5 More User-friendly approaches

One critical discussion would be to aim for more user-friendly approaches and adjust the implementations under such criteria. There will be a few aspects that we can already target to work on after improving the challenge of adjusting the direction of implementing our learning environment.

After receiving a survey, we realized that creating a new course is confusing; therefore, we need to improve it. Expecting a new user suddenly provides the platform with much information about various entities that can make the course creation process vague and

complicated. A standard solution would be asking the user to enter the data step-by-step, in some sort of wizard-style approach.

Figure 51 represents a mock-up of the idea of creating a new course as a multiple-style wizard-style procedure.

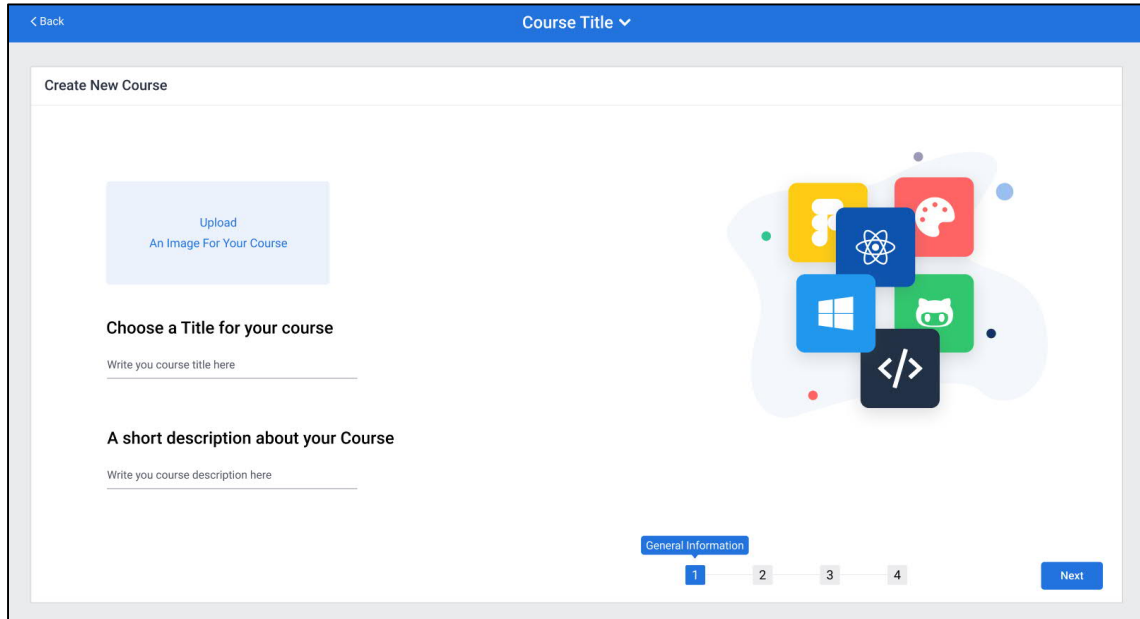


Figure 51: Course Designer; Wizard view

6.3 Future Research Suggestions (What is next?)

Many areas have the potentials of research for more comprehensive findings to help us understand adjusting our implementations in Learning Environments. Understanding users' behavior is by far the most important of them. Also, integrating trending topics such as blockchain, machine learning, AI, data science, etc. would be an exciting possibility now that we have a proper structure to maintain.

In short, here is a list of suggestions for future research:

- Understanding users' behavior in more details by using tools such as Hotjar
- Possibilities of using Machine Learning in more advanced recommendation systems related to the learning environment
- Developing a parallel learning environment in Virtual Reality and unleashing the power of virtual environments
- Global analysis regarding various markets to understand their limitations to offer an ultimate learning solution
- The science of colors and their impact on learning environments
- Using blockchain and decentralized services to provide more secure and reliable educational services
- Real-time collaboration potential in learning environments
- Adoption of social networks strengths and capabilities in educational solutions

7 CONCLUSION

7.1 Community-first approach

In this thesis, I did not discuss whether it is more beneficial to use LMS systems or not. A vast percentage of studies indicates the positive impact of using such systems. Briefly, I am going to represent a research from EDUCAUSE in the following:

Figure 52 represents a study indicating the main ways that the faculties are typically using the LMS. [46, p. 10]

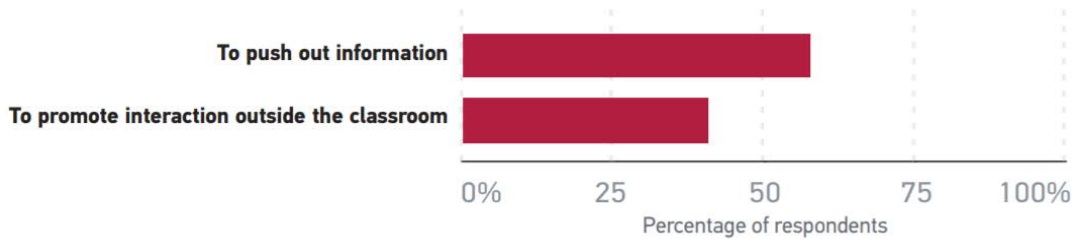


Figure 52: Ways in which faculty typically use the learning management system

And Figure 53 demonstrates Overview of faculty satisfaction with LMS features. [46, p. 12]

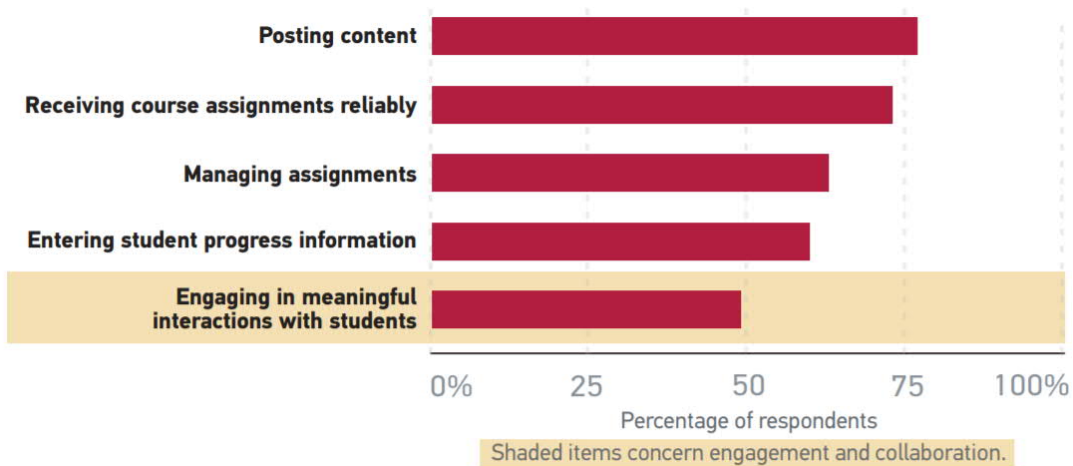


Figure 53: Overview of faculty satisfaction with LMS features

As you may see in the figure, the lowest satisfaction is for engaging in meaningful interaction with students. This is the major reason why we implemented TechClass Smart Online Learning Community to allow the students and lecturers to interact in more meaningful ways. Therefore, introducing the term community-first was an ambitious approach to redirect the focus on such features.

Following our community-first approach, we made a bold and significant move to take a step back, observe, analyze, and finally adjust the direction of our implementations toward a beneficial path. Such a decision automatically fixed a significant amount of our technical debts and made the future features implementations noticeably easier. The current structure will also make it much easier for us to analyze the behavior of our users.

We achieved many values, whether we reached a new starting point. We need to maintain our singular vision mindset and improve it to make TechClass Smart Online Learning Community as a Learning Environment a more efficient and practical platform.

7.2 Limitations and implementation of features

Introducing the very first edition of TechClass Smart Online Learning Community tackled many major issues. In the year 2017, before I analyze and have the IT team implement the new generation of TechClass Learning Environments, I asked a group of students, “What are the benefits of studying as a group?”, and Figure 54 represents the result. [47]

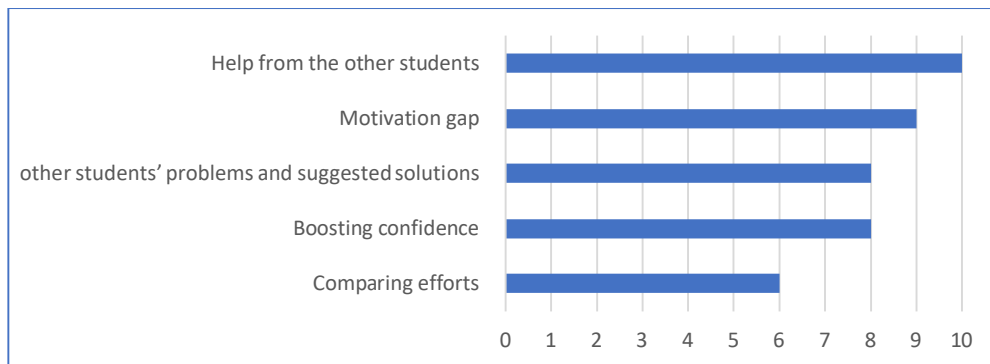


Figure 54: Questionnaire result

Help from the other students (100%):

All students agreed that they can ask other classmates to help them understand a subject while they are learning as a group.

Motivation gap (90%)

Nine out of ten students decided that they would be inspired to learn the content by learning as a group. If someone knows what to do to complete a task, but chooses not to do it, it is seen as a motivation gap. [48]

Reading about the other students' problems and suggested solutions (80%):

Eight out of ten students decided that reading what others are having trouble with will be helpful, whether they know a solution and assist them, or whether they do not know and would read to understand.

Boosting confidence (80%)

Eight out of ten students said that studying as a group will increase their confidence to continue studying.

Comparing efforts (60%)

Six out of ten students decided it would be a nice insight to compare their result with the other members of the group.

As a result of the implementation of the new generation of TechClass Learning Environments, we tackled all the mentioned issues of the previous editions and already provided actual functionalities and features making such concerns possible.

7.2.1 The decision about having a separate Design System

As we mentioned in 4.3.4, we decided to use an entirely made design system with smaller modifications to get back to a more conventional approach. Such a decision is estimated to cost two sprints, and at first, it would be upsetting to leave behind our design system, but various factors validate such a decision.

We got attached to the look and feel of the platform. It is unique and lovable, but the question is the cost of such a difference, mostly from getting attached to a decision without considering the costs and hardships.

Unlike many of our decisions, it does not follow the MVP mindset. We are putting many resources into an area that is not bringing back that much business value. This is a red flag that automatically makes such decisions unjustified.

The team's developers and QAs should do much more challenging work to follow our design system criteria. Nowadays, popular design systems are similar to the open-source community mindset; hundreds or even thousands of skillful people contribute to making them better and better over time. They have finalized and fixed many concerns and issues, which is impossible for a limited designer on our team to compete.

In TechClass, our approach so far was allowing members to experience and making big decisions. However, for the past 2.5 years since we established our IT team, there were situations that we had to go in a different direction, which sometimes was leading to stopping the collaboration with some members.

Consider hiring a designer to make the relayed decisions knowing the team's resources and limitations and align with the industry standards. When there are more prepared solutions to apply and benefit from, creating our solution is similar to re-inventing the wheel. Such an example is similar to create a similar JavaScript library close to React.js, and then the front-end developers create and maintain that. This will act as a black hole consuming the project's resources.

Now that we decide to follow a more common industry standard regarding the design systems, many excellent choices are popular, influential, and a sizeable community maintains them.

A few of the best design systems to use are Material UI by Google, Fluent by Microsoft, Eva Design System by Akveo, Carbon from IBM, Polaris from Shopify, Spectrum by Adobe, Ant Design System, Atlassian, etc.

Having arguments between the design and developer team was a regular part of our work. One of the main reasons was that the mock-ups were not finalized as the developers expected to have such personalized designs. Automatically, many of the details were missing from the final work, which was being delivered to the developer team. It was creating tension from time to time in the daily meetings. A more conventional approach in the design system as a dictionary will not decrease such tension between the design and development teams.

Implementation costs will significantly get faster now that we do not need to take care of the details. Also, React.js resources already implemented the components in React.js for the most popular design systems. Therefore, picking a design system that has such resources is essential.

The platform's design will become more user-friendly for the users, thanks to following a more standard approach.

References

- [1] A. Elayyan, "Toward quality assurance and excellence in higher education", Denmark: River Publishers, 2018.
- [2] J. Sumner, "Serving the System: a critical history of distance education", Open Learning, Vol. 15, No.3, 2000.
- [3] F. Eftekhari, "Developing an Intelligent e-Learning Portal: Challenges and Opportunities," Page 2. 2015. <http://urn.fi/URN:NBN:fi:amk-201605239346> (accessed Oct. 24, 2020).
- [4] Commission of the European Communities, "The E-learning Action Plan: Designing Tomorrow's Education. Commission of the European Communities," 2001.
- [5] N. Fathema, D. Shannon, and M. Ross, "Expanding The Technology Acceptance Model (TAM) to Examine Faculty Use of Learning Management Systems (LMSs) In Higher Education Institutions," in MERLOT Journal of Online Learning and Teaching, June 2015.
- [6] E. Pecheanu and D. Stefanescu, L. Dumitriu. and C. Segal, "Methods to evaluate open source learning platforms," in IEEE Global Engineering Education Conf. (EDUCON), April 2011, pp. 1152-1161.
- [7] A.S. Al-Ajlan, "A comparative study between e-learning features. Methodologies, tools and new developments for e-learning," 2012, pp.191-214.
- [8] N. Cavus and T. Zabadi, "A comparison of open source learning management systems. Procedia-Social and Behavioral Sciences," 2014, pp. 521-526.
- [9] D. Ülker and Y. Yilmaz, "Learning management systems and comparison of open source learning management systems and proprietary learning management systems," Journal of systems integration, 7(2), 2016, pp.18-24.
- [10] Google Classroom Software - 2020 Reviews & Pricing, <https://www.softwareadvice.com/lms/google-classroom-profile/> (accessed Nov. 3, 2020).
- [11] Moodle Software - 2020 Reviews, Pricing & Demo, <https://www.softwareadvice.com/lms/moodle-profile/> (accessed Nov. 3, 2020).
- [12] Docebo LMS Software - 2020 Reviews, Pricing & Demo, <https://www.softwareadvice.com/lms/docebo-lms-profile/> (accessed Nov. 3, 2020).
- [13] Easy LMS Software - 2020 Reviews, Pricing & Demo, <https://www.softwareadvice.com/lms/easy-profile/> (accessed Nov. 3, 2020).
- [14] G Suite Editions | Google for Education, G Suite Editions | Google for Education <https://edu.google.com/products/gsuite-for-education/editions/> (accessed Nov. 3, 2020).
- [15] Pricing | Moodle, <https://moodle.com/pricing/> (accessed Nov. 3, 2020).
- [16] Docebo LMS Pricing – Let's Keep Pricing Simple <https://www.docebo.com/lms-elearning-platform-docebo-prices/> (accessed Nov. 3, 2020).
- [17] LMS pricing | Easy LMS, <https://price.easy-lms.com/start-trial/> (accessed Nov. 3, 2020).
- [18] F. Eftekhari, "TechClass Smart Online Learning Community: Core", in INTED2019 educational Conf. proceedings, pp. 2893-2901.
- [19] F. Eftekhari, "TechClass Smart Online Learning Community: Community Features" in INTED2019 educational Conf. proceedings, pp. 2893-2901.
- [20] L. McCoy, Microsoft Azure Explained: what it is and how to use it | CCB Technology, <https://ccbtechnology.com/what-microsoft-azure-is-and-why-it-matters/> (accessed Nov. 3, 2020).
- [21] JSON API – Latest Specification (v1.0), <http://jsonapi.org/format/> (accessed Oct. 24, 2020).

- [22] M. Anastasov, CI/CD Pipeline: A Gentle Introduction – Semaphore, <https://semaphoreci.com/blog/cicd-pipeline> (accessed Nov. 3, 2020).
- [23] Docker overview | Docker Documentation, <https://docs.docker.com/get-started/overview/> (accessed Nov. 3, 2020).
- [24] K. Bracey, What Is Figma?, <https://webdesign.tutsplus.com/articles/what-is-figma--cms-32272> (accessed Nov. 3, 2020).
- [25] E. Ries, *The Lean Startup*, New York, USA: Currency International Edition, 2011.
- [26] A. Hacq, Everything you need to know about Design Systems, <https://uxdesign.cc/everything-you-need-to-know-about-design-systems-54b109851969> (accessed Nov. 3, 2020).
- [27] Git, <https://git-scm.com/> (accessed Nov. 3, 2020).
- [28] G. Levin, An API-First Development Approach, <https://dzone.com/articles/an-api-first-development-approach-1> (accessed Oct. 24, 2020).
- [29] Introduction to ASP.NET Core, <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-3.1> (accessed Nov. 3, 2020).
- [30] What is REST - REST API Tutorial, <https://restfulapi.net/> (accessed Nov. 3, 2020).
- [31] Why Cypress? | Cypress Documentation, <https://docs.cypress.io/guides/overview/why-cypress.html> (accessed Nov. 3, 2020).
- [32] React Official Documentations, <https://reactjs.org> (accessed Nov. 3, 2020).
- [33] TypeScript Official Documentations, <https://www.typescriptlang.org> (accessed Nov. 3, 2020).
- [34] E. Politou, E. Alepis, C. Patsakis, “Forgetting personal data and revoking consent under the GDPR: Challenges and proposed solutions,” *Journal of Cybersecurity*, Volume 4, Issue 1, 2018.
- [35] A. Prokopiško, Software development life-cycle (SDLC), <https://medium.com/@artjoms/software-development-life-cycle-sdlc-6155dbfe3cbc> (accessed Oct. 24, 2020).
- [36] J. Gharibshah, Z. Gharibshah, E. Papalexakis, and M. Faloutsos, “An Empirical Study of Malicious Threads in Security Forums.” in *Companion Proceedings of World Wide Web Conf. Association for Computing Machinery*, New York, NY, USA, 2019, pp. 176–182.
- [37] N. Young, What is wireframing | Experience UX, <https://www.experienceux.co.uk/faqs/what-is-wireframing/> (accessed Nov. 3, 2020).
- [38] Introduction – Slate, <https://docs.slatejs.org/> (accessed Nov. 3, 2020).
- [39] Smoke Testing - SOFTWARE TESTING Fundamentals, <https://software-testingfundamentals.com/smoke-testing> (accessed Nov. 3, 2020).
- [40] H. Shah, What is Functional Testing? Explained with Test Cases & Example, <https://www.simform.com/functional-testing/> (accessed Nov. 3, 2020).
- [41] What is Regression Testing? Definition, Tools, Method, and Example, <https://www.softwaretestinghelp.com/regression-testing-tools-and-methods/> (accessed Nov. 3, 2020).
- [42] Integration Testing - SOFTWARE TESTING Fundamentals, <https://software-testingfundamentals.com/integration-testing/> (accessed Nov. 3, 2020).
- [43] C. Kade, Introduction to Front-End unit testing, <https://dev.to/christopher-kade/introduction-to-front-end-unit-testing-510n> (accessed Nov. 3, 2020).

- [44] What is Microsoft Teams and who should be using it? | Compete366, <https://www.compete366.com/blog-posts/microsoft-teams-what-is-it-and-should-we-be-using-it/> (accessed Nov. 3, 2020).
- [45] G. Hollnagel, "The Benefits and Limitations of Using Online Technology in Education: Students' and Instructors' Perceptions on Usefulness and Task-technology-fit", Dresden, Germany, pp. 6.
- [46] E. Dahlstrom, D. Brooks, and J. Bichsel, "The Current Ecosystem of Learning Management Systems in Higher Education: Student, Faculty, and IT Perspectives," EDUCAUSE, Washington, USA, 2014.
- [47] F. Eftekhari, "Progressing from an eLearning portal to a smart online learning community," INTED2018 Proceedings, pp. 1997-2005, 2018.
- [48] J. Dirksen, "Where do we start" in Design for how people learn, Page 8., Berkeley: New Riders, 2016.

A. Questionnaire

[Student – Lecturer - Staff]

1. How easy is the learning environment to use?

Very easy to use - Easy to use - Neither easy nor difficult to use - Difficult to use - Very difficult to use

Note: This question is to investigate the usability of the product.

[Student – Lecturer - Staff]

2. How satisfied are you with the look and feel of the learning environment?

Extremely satisfied - Very satisfied - Somewhat satisfied - Not so satisfied - Not at all satisfied

Note: This question gives a good insight about the UI/UX of the product.

[Student – Lecturer - Staff]

3. How satisfied are you with the performance, loading, and responsiveness of the learning environment?

Extremely satisfied - Very satisfied - Somewhat satisfied - Not so satisfied - Not at all satisfied

Note: This question focuses on the performance, responsiveness, and related affairs of the product.

[Student – Lecturer - Staff]

4. How well does the learning environment meet your needs?

Extremely well - Very well - Somewhat well - Not so well - Not at all well

Note: This question delivers a general overview of the product.

[Student – Lecturer - Staff]

5. Which of the following words would you use to describe the learning environment? Select all that apply.

A1) Reliable – B1) High quality – C1) Useful – D) Unique – C2) Impractical – B2) Poor quality – A2) Unreliable

Note: This question delivered various aspects that the users understand from the characteristics of the product.

[Student – Lecturer - Staff]

6. How likely is it that you could recommend the learning environment to a friend or colleague?

Very unlikely 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 Very likely

Note: This question shows user feedback directly, and it is also a popular way of measuring customer satisfaction. If a user does not recommend a product to their friend, they may feel disappointed in some aspects.

B. Data Models

At first, we needed to reimagine our learning environment to include innovative ideas and create new possibilities. Second, we needed to combine such innovation with what we already provide and then aim for an outcome, which fulfills both goals.

The first step was to list the various models as entities we had to assure, we are picturing areas to include such data.

Although we can list all the following models under Course Models, we divided them into the following four categories to explain them more easily.

Course Models

Course Models represent the general data of the courses. Table 1 describes the related models.

Table 1: Course model and its properties

| # | Model | Important Properties |
|---|-----------------|--|
| 1 | Course | Id – Title – Description – CoverObjectId – DeadlineDays – Credit – Course CategoryId, CompletePercentage |
| 2 | CourseCategory | Id - Name |
| 3 | CourseLecturer | Id – CourseId - LecturerId |
| 4 | CourseProgramme | Id – CourseId - ProgrammId |

Content Models

Content Models represent the data related to the content of the courses. Table 2 explains the related models.

Table 2: Content mode and its properties

| # | Model | Important Properties |
|---|---------|---|
| 1 | Chapter | Id – Title – Order - CourseId |
| 2 | Section | Id – Title – DescriptionObjectId – Order – ChapterId - Type |

Task Models

Task Models represent the task categories' data such as assignments, quizzes, projects, and essays. Table 3 justifies the related models.

Table 3: Task model and its properties

| # | Model | Important Properties |
|---|--------------------------------|---|
| 1 | AssessmentRequest | Id – RequestId - EnrollmentId |
| 2 | Assignment | Id – Point – DescriptionObjectId – IsMandatory – SectionId – Type - Title |
| 3 | AssignmentEnrollment | Id – Score – EnrollmentId – AssignmentId – DescriptionObjectId - Feedback |
| 4 | AssignmentEnrollmentAttachment | Id – AttachmentObjectId – AssignmentEnrollmentId - FileName |
| 5 | MultiChoiceAnswer | Id – Title – QuestionId – IsRightChoice - Order |
| 6 | Question | Id – Title – QuizId – Score - Order |
| 7 | QuestionEnrollment | Id – QuizEnrollmentId – QuestionId - MultiChoiceAnswerId |
| 8 | Quiz | Id – DescriptionObjectId – IsMandatory – SectionId – AnswerTime – PassScorePercentage – Point - Title |
| 9 | QuizEnrollment | Id – Score – QuizId – EnrollmentId – ScorePercentage - TotalScore |

Enrollment Models

Enrollment Models represent information about the enrollment of the students. Table 4 explains the related models.

Table 4: Enrollment model and its properties

| # | Model | Important Properties |
|---|------------|---|
| 1 | Enrollment | Id – ExpirationTime – CompletionTime – DeadlineDate – Enrollmenttime – Score – Status – AssessmentStatus – StudentId – ProgrammId – CourseId – LastSectionId – Feedback - CertificateId |