Bacheloroppgave

IBE600 IT

Hybrid Learning Management System built on Craft CMS

Andreas Ringstad og Sander Holmen

Totalt antall sider inkludert forsiden: 45

Totalt antall ord: 8508

Molde, 31. mai 2022

Høgskolen i Molde
Vitenskapelig høgskole i logistikk

Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. Manglende erklæring fritar ikke studentene fra sitt ansvar.

Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:		
1.	**Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.**	☒
2.	**Jeg/vi erklærer videre at denne besvarelsen:** • ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands. • ikke refererer til andres arbeid uten at det er oppgitt. • ikke refererer til eget tidligere arbeid uten at det er oppgitt. • har alle referansene oppgitt i litteraturlisten. • ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.	☒
3.	**Jeg/vi er kjent med at brudd på ovennevnte er å betrakte som fusk og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. Universitets- og høgskoleloven §§4-7 og 4-8 og Forskrift om eksamen §§14 og 15.**	☒
4.	**Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i URKUND, se Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver**	☒
5.	**Jeg/vi er kjent med at høgskolen vil behandle alle saker hvor det foreligger mistanke om fusk etter høgskolens retningslinjer for behandling av saker om fusk**	☒
6.	**Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider**	☒

Personvern

Personopplysningsloven

Forskningsprosjekt som innebærer behandling av personopplysninger iht. Personopplysningsloven skal meldes til Norsk senter for forskningsdata, NSD, for vurdering.

Har oppgaven vært vurdert av NSD? □**ja** ☒**nei**

- Hvis ja:

Referansenummer:

- Hvis nei:

Jeg/vi erklærer at oppgaven ikke omfattes av Personopplysningsloven: ☒

Helseforskningsloven

Dersom prosjektet faller inn under Helseforskningsloven, skal det også søkes om forhåndsgodkjenning fra Regionale komiteer for medisinsk og helsefaglig forskningsetikk, REK, i din region.

Har oppgaven vært til behandling hos REK? □**ja** ☒**nei**

- Hvis ja:

Referansenummer:

Publiseringsavtale

Studiepoeng: 15
Veileder: Bård-Inge Austigard Pettersen & Per Kristian Rekdal

Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten (Åndsverkloven. §2).
Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage HiM med forfatter(ne)s godkjennelse.
Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved Høgskolen i Molde en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering: ☒**ja** □**nei**

Er oppgaven båndlagt (konfidensiell)? □**ja** ☒**nei**
(Båndleggingsavtale må fylles ut)
- Hvis ja:
Kan oppgaven publiseres når båndleggingsperioden er over? □**ja** □**nei**

Dato:

Preface

Working on this thesis has been a new and exciting experience. We wish to thank our bachelor supervisors Bård-Inge Austigard Pettersen and Per Kristian Rekdal for all the time and encouragement they have given us as we have tried to do their vision justice.

Months prior to starting our bachelor's thesis Bård-Inge and Per Kristian introduced us to the idea of an education system they wish already existed. The system would equal the importance of how teachers create courses with the importance of how the students attend courses, would make it a lot easier to share content between educational institutions, and would give different students that take the same courses a different experience through adaptability.

For our thesis they suggested that we try to design and describe this system and possibly try to build parts of it. It is a big and complex system, so even though we imagined we would have time to build some of it, the process of designing it was time consuming and challenging enough for us.

Abstract

Education has been and always will be changing. These last few years we have seen a massive rise of digital tools used in education, especially during the Covid-19 pandemic. This has introduced a new type of learning called “hybrid learning”. This is the term explaining education when there are both students physically on campus taking a course, as well as online students located anywhere taking the same course. The challenge with this is that there are no learning management systems (LMS) that are suited for this kind of teaching.

In this report we are designing an LMS that can help this type of teaching, as we believe hybrid learning is the future of education. We also want to make a better LMS by introducing new features that have not existed in these types of systems before, like sharing of course material, incentives for creating course material, and adaptive learning on all levels of education. The design process will be based on Craft CMS (content management system) which we have chosen to build the LMS on top of. We look at which of the features we want Craft can help us with. And look at how we can achieve the features we want that Craft does not offer, by extending the system with plugins and modules.

Contents

1.0 Introduction ... 1

1.1 Purpose ... 1

1.2 Why ... 1

1.3 What we are making ... 2

1.4 Scope ... 2

2.0 Literature review ... 4

3.0 Case description ... 6

4.0 Requirements ... 8

4.1 Learning management systems ... 8

5.0 Design ... 11

5.1 System architecture ... 11

5.1.1 xBlocks ... 11

5.1.2 Learning paths ... 12

5.1.3 Bookstore ... 14

5.1.4 Institutions’ Client ... 14

5.2 Shareability ... 15

5.3 Rating ... 15

5.4 Adaptability ... 16

5.5 Analytics ... 17

5.5.1 Analytics for subscription/monetizing ... 17

5.5.2 Analytics for student progress ... 18

5.6 Live communication ... 18

5.7 Database ... 18

6.0 Integrating with Craft ... 21

6.1 What is Craft CMS? ... 21

6.1.1 Features of Craft ... 21

6.2 Extending Craft ... 24

6.3 Plugins ... 25

6.4 Modules ... 29

7.0 Business model ... 32

8.0 Conclusion ... 33

9.0 Future Work ... 34

References 35

Acronyms 37

1.0 Introduction

Learning management systems (LMS) can typically be sorted into two categories. Either it is an institution specific LMS like Canvas, itsLearning or Blackboard, and is locked to the students attending that institution. Or it is an open-source type LMS, like Open edX or Coursera, which are universal systems open to everyone. This second type is often very large and complex and even though they can be used by institutions they can be quite difficult to implement.

Our goal is to design a system that is something in between those two categories. A system which can host hybrid type courses, which can easily be implemented at one institution, and which can share material for building courses between all institutions. The system will be a hybrid solution that works well for both online courses and campus courses. It will also facilitate great course adaptivity.

1.1 Purpose

The reason for designing a system like this is after experiencing today's learning management systems both from a student and teacher perspective, we believe that the systems that exist do not fulfill the needs of modern, digital education, which has gained massive momentum during the Covid-19 pandemic (Regjeringen 2021).

1.2 Why

During the Covid-19 pandemic, schools were forced to implement digital lectures. What students experienced was that the digital teaching did not have the same quality as when they had physical lectures on campus. At the same time the students appreciated the flexibility introduced by being able to study where and when they wanted with digital lectures. Now that the pandemic is coming to a potential end, and students return to campus, it is a golden opportunity to use physical and digital teaching intertwined where they are best suited. Digital teaching should now be used to improve the quality of education. Digital technologies have potential to move the courses, education, and research forward. It should be used to develop more customized and flexible courses with high quality (Regjeringen 2021).

1.3 What we are making

Something that we want to improve with our system is the ability for teachers to create adaptive courses. By adaptivity we mean the ability to give different content to different types of students in the same course. With the existing systems it is possible to do, but it is either very complicated, hard, and requires great technical skill, or it simply takes a lot of time to do. We want to simplify this experience for teachers. We believe this can lead to a better experience both for learners and for teachers, as it gives the opportunity to create different learning paths for different groups of students within the same course.

An example could be IT students and logistics students at Molde University College. Both take the same course called “Statistics 1” which is an introduction to statistics. These students have different background and knowledge when they start the course. To create the best possible experience and education for these two groups, a teacher could, with our system, make two different learning paths or potentially branches of learning paths, which we will come back to later. One of the paths could use programming to solve the problems and teach in a way that is tailored for those with prior knowledge in IT. While the other path is tailored for those with knowledge in logistics. And this goes on for all types of groups that take the course. In addition, as teachers skip over the basic concepts that students should already be familiar with, adaptivity can be used to include explanations of these concepts either through a link or a drop-down section. This would help the students that would benefit from revisiting it, while being easily dismissible to those that do not.

Another feature we want to add to our system, that is not widespread among existing learning management systems, is sharing of course material. By sharing we mean both internal sharing within an institution, and external sharing with other institutions. This is to make it easier for teachers to create good courses, as they can use existing material that is proved to be of good quality and implement it into their courses. This will benefit both the teachers and the students.

1.4 Scope

This report will talk about the ideas we have and the design decisions we make for the LMS system we want to create. We will be explaining why we make the choices we do, and why we believe they will solve the challenges we are facing. There will be no programming, other than some code to show how one could go about using Craft to build something big that goes beyond the built-in features, like we do.

In the case description we will look at education today, and how we came up with the idea to design this type of system. Then in the requirements section we will talk about how different learning management systems look today, and what we can take from them, as well as which new features we are going to include in our system. Then we will start the design phase. Here we will look at the system architecture and why we make the design decision we do. Then we show how our design can be implemented using Craft CMS, and we also look at how we can extend the system to include the features that Craft cannot help us with. Next, we will take a look at the business model for how the content creators will get paid. And finally, we talk about what would be the next steps in order to bring this system to life.

2.0 Literature review

Learning Management Systems (LMS) provide teachers and students with an online classroom environment. A modern LMS reinforces the education process with structures that promote online collaboration, discussions, and communication among other LMS users (Bradley 2021). A LMS can be described as a structured repository of courseware materials provided with additional communication facilities (Scerbakov 2019).

An effective LMS is a vital component of a university with an overall E-learning structure (Scerbakov 2019). We know that higher education institutions are more interested in using adaptive learning when teaching. Adaptive learning systems can be everything from simple systems based on a set of rules to complex systems with artificial intelligence (Mirata 2020). In this type of learning environment, the system should be adapted to the learner's need to some extent (Normadhi 2019).

A modern LMS ought to have adaptive functionality because this can reduce the time it takes for learners to become proficient, eliminating the need for them to cover content they already know. To enable this, content must be modified into smaller objects. Future success of adaptive learning is more about the design of content, not technology (Ramani 2017).

Any learner would benefit from an adaptive model in learning, but there is one group that gains more, and that is the under-achievers. Those who usually don't succeed in education can now meet their goals with help from the supportive nature of the adaptive system (Serhat 2021).

Different types of learning management systems have a variety of features and combinations. Some of the types are installed LMS, web based LMS, hosted LMS, SaaS LMS, open-source LMS, and commercial LMS (Kunju 2021).

Installed LMS is the classic, old way of installing applications on a computer. This is usually an expensive solution because it requires maintenance and its own IT staff (Kunju 2021).

Web based LMS works mostly through your browser or web-based solution through the internet. The software vendor manages the system and updates, improves, and maintains the application continuously. This is usually cheaper than the installed solution (Kunju 2021).

Hosted LMS and SaaS LMS are both hosted on the cloud. The difference is the maintenance of the infrastructure. In the hosted solution the institution is responsible for hosting and controlling the uptime and security of the server. Updates and upgrades are managed by the institution’s IT team or technical associates. In the SaaS application the responsibility of the service lays with the vendor. The institution only pays for the system usage (Kunju 2021).

Any of these types can be open-source, or commercial. The benefits of an open-source platform are that it is available for free, especially its basic versions. It can be scaled to a large number of users without additional cost. The flexibility, as you can customize the design and features as you need. The risks of open source can be that they are not very user-friendly. In fact, they can be very difficult to use, especially for beginners. The user interface can also be very complex which makes it difficult to administer. Also, when an LMS is open source, they often don’t come with dedicated technical support. Most of the support is forum based, which makes it more difficult to find solutions in a short time frame (Polakowski 2021).

Today there exists no optimal LMS that offers a package with all-in-one features for the online educational process (Salah 2021).

For IT systems used in higher education it is important that they are simple, clear and have standardized interfaces. That the educational data is standardized and harmonized. Data should also be created once and then managed in a source and reused (Regjeringen 2017).

3.0 Case description

To understand what we are trying to solve, we must explain four concepts: physical teaching, digital teaching, blended teaching, and hybrid teaching.

Physical teaching is the concept where all students are present at the institution. The lectures can take place in a classroom, an auditorium, a group room, a laboratory, in the field or in practice. Digital tools can be used to support the teaching when appropriate (Regjeringen 2021).

Digital teaching is when all students participate digitally. The teaching can consist of video lectures, online based group work, and other digital tasks to mention some. The lectures can occur in real time or be watched by students at any time (Regjeringen 2021).

Blended teaching is when the teaching consists of both digital and physical lectures, but they take place separately. The physical and digital arenas are chosen based on what is most suitable. Principles like “flipped classroom” can be used. The course may be offered as digital lectures, physical group work, and physical or digital tasks (Regjeringen 2021).

Hybrid teaching is when some students are present physically, and some students participate online at the same time. This way of teaching requires technological equipment and knowledge, as well as pedagogically justified arrangements that gives all students an equal opportunity to participate and achieve learning (Regjeringen 2021).

We want to design a system which is suited for hybrid teaching, which we believe is the future of education. To achieve this, the system must make it as easy as possible for the teacher to create courses that can be used by a variety of students, both online and on campus. The system also needs to facilitate learning for all types of students, i.e., young, and old, full time, and part time etc.

Another aspect of our learning management system is that we want institutions, publishers, and teachers, to be able to share educational content with each other. This we believe could lead to networking between institutions and teachers as they could cooperate on creating content. Cooperation like this could lead to improved educational content for the students.

The graphical user interface must be user-friendly and simple to minimize the time teachers and students use to navigate and interact with the system. Another important feature of this LMS is the adaptive functions for when one creates a course. We want the teachers to be able to customize courses at a very low level, without it being a very complicated task.

The issue is that there exists no learning management system that is optimal for hybrid learning. We want to see if we can take features from different existing systems, as well as adding a few completely new features, to make a new LMS that suits the modern, hybrid teaching.

4.0 Requirements

Before we can start designing our system, we need to know what we are building, what it requires and needs. Because we are making something that exists in some way before, we have to analyze these existing systems, and see which of the existing features we want to take with us, which we do not want, and which new features we must implement.

4.1 Learning management systems

When attempting to make a system that is better than what already exists there are a lot of features that the system needs. We will look at features of normal LMS and features of MOOC oriented LMS.

Popular features of a standard LMS (Lynn 2021):

- Course management – must be able to easily deliver course content to the learners.
- Branding – the institution having its own visual identity will help make a better environment for the learners.
- Integrations – an LMS that integrates with software can automate some tasks and reduce the amount time-consuming tasks.
- Automation – automation that removes repetitive tasks like notifications will help reduce time spent on admin tasks.
- Exams – test your learners and measure their progress.
- Surveys – feedback and evaluating the student’s experience
- Gamification – creates a more engaging learning experience.
- Configurable roles and permissions – define structure and hierarchy.
- Groups – collection of people with common interactions.
- Forum – places for users to discuss and interact with each other.
- Analytics – to showcase data of learner’s progression and more.
- Learning paths – some type of order the learner is supposed to go through the course material.
- Mobile – accessible on mobile phones and tablets.
- Notifications & reminders.
- Localization – multi language support and cultural adaptability.

- Security.
- Single sign-on.
- Messages – any user can send a message to any other inside the organization.
- Curriculum – the source material the courses are built on.

Most of these features we will want to include in our new LMS.

MOOC (Massive Open Online Courses) is a type of course hosted on an LMS. Therefore, a MOOC itself doesn't need any features, but the LMS it is hosted on does need the right features to be capable of doing that.

Characteristics of a MOOC platform (Bates 2019):

- Massive – a MOOC platform in theory has infinite scalability because the cost of adding extra participants is basically zero for the institution offering the MOOC.
- Open – the only pre-requisite for participants is that they have access to a computer or a mobile device and the internet. Initially, the access to MOOCs is free of charge, although some institutions have started charging a fee for assessment leading to a badge or certificate.
- Online – mostly, MOOCs are offered holly online, and not as hybrid format to use on campus.
- Courses – they are organized into a whole course, and when you finish a course, you may receive a badge or a certificate.

  As well as the characteristics explained by the letters "MOOC", one more thing differs it from standard LMS systems, and that is it may have subscription and purchase features to access some courses.

In addition to the features of a standard LMS, there are some other features that we would like to see to fulfill our vision of how an LMS should be:

- Shareability – a library created by someone at one institution should be shareable so it can be used by other courses at other institutions if the creator chooses to do so.
- Adaptability – the possibility that students taking the same course are presented with different content based on some user properties.
- Live communication – the ability to have live meetings and thereunder potentially live classes.
- Bookstore – a marketplace where publishers can add books.

5.0 Design

In this part we will describe the design and architecture we have planned for the system. We will also show some images to better illustrate what we mean in the text.

5.1 System architecture

We plan to build a system where each institutional organization gets their own website and domain, or client, which they can customize to their liking. The rest of the system will be the same for all. The courses are specific and locked to each institution, but everything else can be shared and reused. The content created by teachers will be put in xBlocks and libraries, which can be used to create learning paths. The system will also allow publishers to publish books digitally that can be used by the teachers as course material.

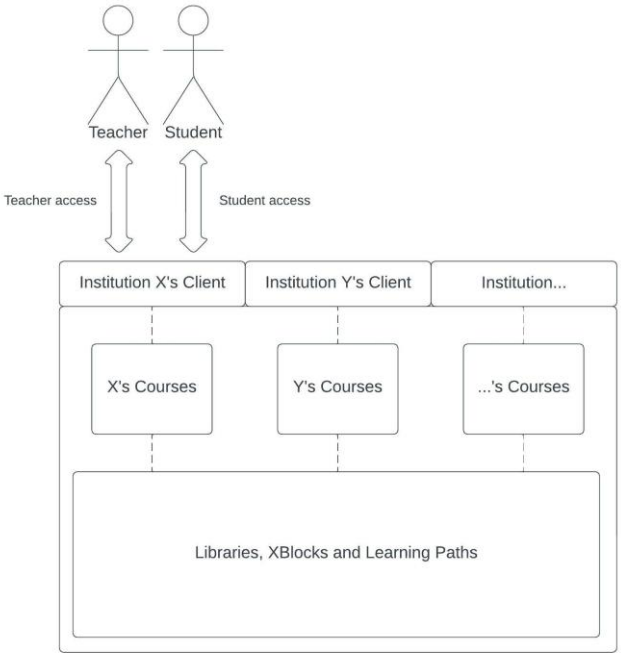


Figure 1: Illustration of the system architecture in our LMS.

5.1.1 xBlocks

The most fundamental part of our solution is the xBlock. xBlock is what we call a small unit of learning material and is what courses consists of. The xBlocks we have focused on initially are text, image, video, and a question xBlock with up to 4 answers to choose from, as they are the most fundamental to get up and running. xBlocks for videos and pictures stores an URL to a server, so the teachers could upload files themselves or use existing pictures and videos like videos on YouTube if they have the right of use. These are some examples of xBlocks, but an xBlock could be anything a teacher wants to use in their courses. As times goes on, we would make more types of xBlocks ourselves, but the plugin system allows other interested parties to create their own, which can then be installed by us with a few clicks. Over time this would increase the flexibility of the teaching material and courses the teachers could create.

When a teacher wants to start creating xBlocks they would first have to create a library to put them in. What the teachers chooses to fill their library with is up to them, but we imagine most libraries would have a theme or topic to serve a more specific purpose. xBlocks are independent from each other, and it should be possible to present them in any order. Teachers can give access to their libraries to other teachers so they can work together and cooperate on creating content for it.

When the teachers are satisfied with their libraries, they can choose whether their libraries are public for other teachers. If the libraries are made public, other teachers can use xBlocks from those libraries in their courses.

5.1.2 Learning paths

When a teacher has made a few xBlocks the next step would be to start making learning paths. A learning path is what the students will see and interact with when they work with course material. The teacher builds the learning path by taking xBlocks from libraries and arranges them in an order pedagogically in line with what the students should learn. These learning paths can be as short or long as the teachers want, but we imagine that the material the course will cover in a week be made into either one or a few learning paths.

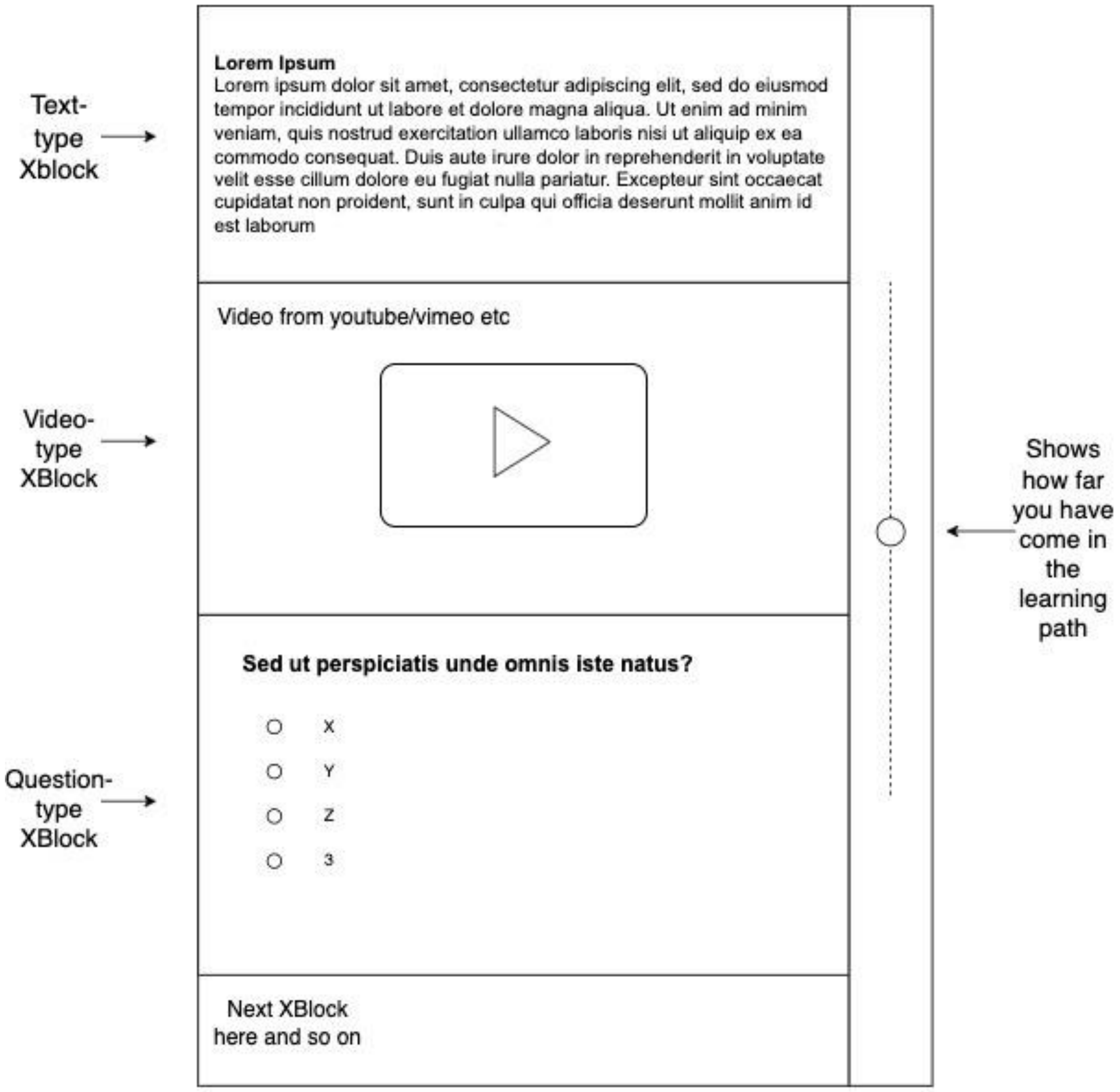

Text-type Xblock
Lorem Ipsum
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum
Video-type XBlock
Video from youtube/vimeo etc
Shows how far you have come in the learning path
Question-type XBlock
Sed ut perspiciatis unde omnis iste natus?
○ X
○ Y
○ Z
○ 3
Next XBlock here and so on

Figure 2: Illustration of how a learning path may look like from a student perspective.

When teachers create an xBlocks they can choose whether the xBlock is free or not for students. In traditional teaching the students get a lot of free course content through the teacher’s lectures. If the students want additional information, they can spend money and buy additional books. With learning paths there will be three options for the students. The learning paths will contain a mix of free and paid xBlocks. The students could choose not to pay anything and just have access to the free xBlocks. This might be limited, so through one subscription the students get access to all the paid xBlocks across every course they have. If this is not enough the students can still purchase the books the courses are based on. With a subscription the price of these books will be discounted.

5.1.3 Bookstore

The bookstore will be an important part of our system, along with the libraries containing xBlocks. The reason we will call it a bookstore is because we want publishers to publish their books there as xBlocks. That means the publishers themselves must split their books into suiting xBlock content before adding them to the system. Then when a teacher wants to build a learning path, they can, as well as using xBlocks made by other teachers, use book type xBlocks made by publishers. The xBlocks from publisher will be hidden behind the same subscription as normal xBlocks.

We see many benefits with this solution. It will lower the amount of money students have to use to buy course material needed for courses, because now everything is under the same subscription. It will also help publishers as it gives them an opportunity to join the digital transformation of education and earn their fair share.

The downside of transforming what used to be physical books into what will be pure digital subscriptions is that the students will not own those books after they have finished their grade. Whether this is something that needs to be solved because most students want to keep their books forever, we do not know. For now, books are sold in physical form, so students have an option to buy the book as well.

5.1.4 Institutions’ Client

Every institution using our system will get their own website which can be customized to their liking. This is the part of the system students see when they are taking courses and interacting with the system. This is also what the teachers see when they preview the content they are working on. However, when the teachers are not previewing, they will interact with the content management part of the system. Here the teachers can also see analytics and see some information about their students.

Every institution also gets one or more admin users. These users are responsible for doing all the overhead work like creating users for to teachers and students, and for giving students access to the courses they are taking.

5.2 Shareability

An important feature of our system will be the ability to share content among institutions and people. All the xBlocks will be stored in libraries owned by users. When a teacher has made their library public, it is visible to other teachers who can then access it and use xBlocks from it in their learning paths and courses.

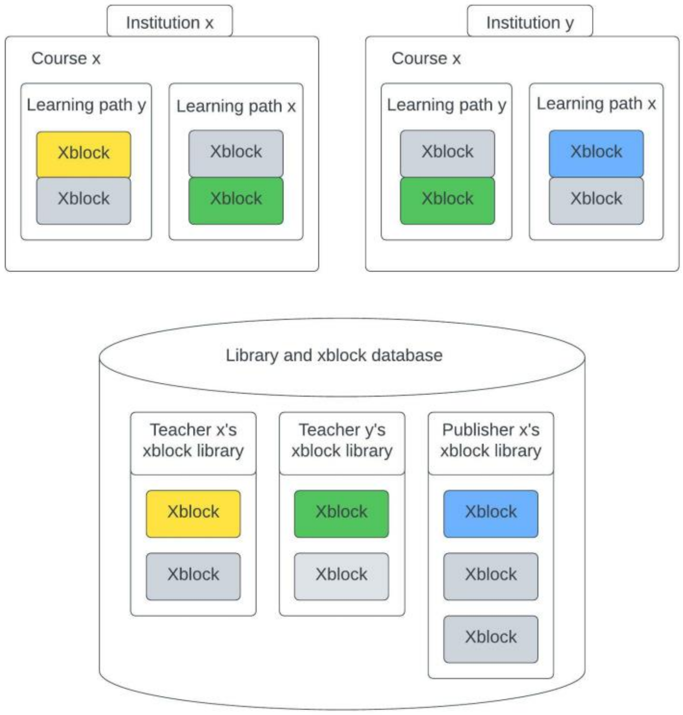


Figure 3: Illustration of how the same course material can be used by multiple institutions.

The ability to share content will save the teachers that either do not want to or do not have time to create the content themselves. And as mentioned earlier this will hopefully increase the quality of the education as education cannot really be better than its content.

5.3 Rating

As more and more teachers and publishers create and publish libraries, rating becomes an important feature. When teachers use libraries other than their own, they can rate them based on their content. Over time as the number of libraries increases the ratings will help teachers find the best content as it rises to the top. This again incentivizes everyone to make the best course material possible as the most popular content has potential to earn the most money.

5.4 Adaptability

One of the features that we are trying to include in our system is adaptability. When teachers have created their learning paths with xBlocks, they can build a course with these learning paths. Learning paths could be arranged in a traditional way, linearly, meaning that students for example have one learning path every week, and therefore they move through all of them one by one in a specified order. Alternatively, the teacher can arrange the learning paths in a tree-like structure, where the students are either handed a branch or choose a branch themselves. Like you can see in the figure below, we have demonstrated two ways this can be used, but ultimately it is up to the teachers if and how they chose to use it.

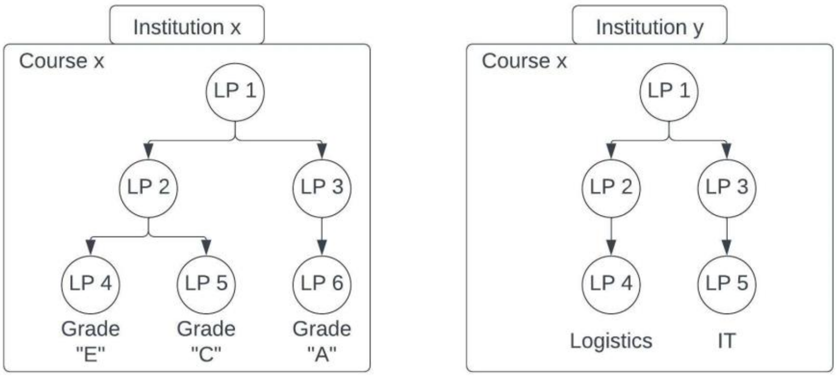


Figure 4: This figure shows two examples of ways to structure learning paths into a course.

The first and probably most useful way is to have branches that represents the different grades you can end up with. A normal way to take a course is to try to learn everything as a course progresses and then accept the grade they end up with as a token to how well they managed to learn everything in that course. This can however be an overwhelming and tiering way to progress through a course, especially for those who might fall behind or fail to learn some of the topics of the course. For the students that ahead of time know that A or B is out of reach, it could be more meaningful and feel more accomplishable to take the C path, and not have to spend energy and time try to understand the A- and B-grade topics.

The other example, like we explained earlier with the course in statistics, is to give different type of students taking the same course an adapted approach to the course more in line with the other courses that student previously have taken.

The first example gave the students the ability to choose the branch they wanted to take themselves. It might not make sense to give the students who have very limited insight into the difficulty of the course the choice of which branch fits them best. Therefore, by surveying the students at the beginning of the semester, the teacher could figure out how quickly and how well they understand some of the topics and then recommend a branch based on that.

Surveying or testing the students at the start of a course could also be used in a less traditional way to have branches that skips parts of the course based on the student's prior knowledge. This could either demand less time of the students or could allow the students to get a deeper understanding of the more advanced topics by spending more time on and explaining them further.

5.5 Analytics

The analytics feature of this LMS we believe will be even more important than it has been with previous systems. With these new ways of structuring courses and teaching we believe an important feature will be the possibility for teachers to see statistics and analytics from their students, courses, and the material they have created.

5.5.1 Analytics for subscription/monetizing

This type of analysis is for the teachers and publishers to see how well their content is doing on the market. For example, if a publisher has several books on a specific topic, they can see which ones are being used the most to create course material. This can potentially help them get insight into what works when it comes to splitting their books into xBlocks. For regular teachers that are creating content, they can see which material is being used the most.

5.5.2 Analytics for student progress

We want an analysis dashboard for viewing data and statistics on how students are doing in the different courses, and on lower levels like in the different learning paths, and even on xBlock level. This will help teachers with a basis to make decisions on how to improve courses and structures.

Some statistics that can be of value for teachers:

- Time used to get through each learning path
- Percentage of correct answers on question xBlocks
- Which learning path different students choose
- Which learning paths lead to the best scores on tests

5.6 Live communication

To make this a hybrid learning platform, there must be an option to host both online and on-campus lectures in the same courses. To solve this, we want to implement Discord in our system. Discord can be described as a chat app in the same category as Skype, TeamSpeak, and Slack. It has many features that make it stand out compared to other communication platforms, and has made it a semipublic, forum-style community platform, embraced by users all over the world (Hornshaw 2021).

When you create a Discord server, it can be configured in many ways. You can set up different roles for members, such as administrators and moderators and other access permissions. You can also reward members for being active on the server (Hornshaw 2021). Because of this we believe Discord will be the best solution for implementing live communication in our system.

5.7 Database

The database is a very vital part of our system as it is fundamental for data storage. An LMS requires a lot of stored information to be able to serve its purpose. It stores information on all the teachers and students. And all the courses, libraries and xBlocks that teachers create. All of this takes a significant amount of space, but the part that takes up the most space is probably student interacting with courses. A course consists of a lot of text, some pictures, and some videos. The pictures and videos are stored outside the database, so they will not take up any space. Any xBlock consisting of text are stored in the database, but text does not require a lot of space. For questions and similar xBlocks that are used in courses you need to store the answer of every student. For a single course this will be the number of questions and number of students squared. So even though text takes up a relative low amount of space, the number of total student answers still makes it a lot.

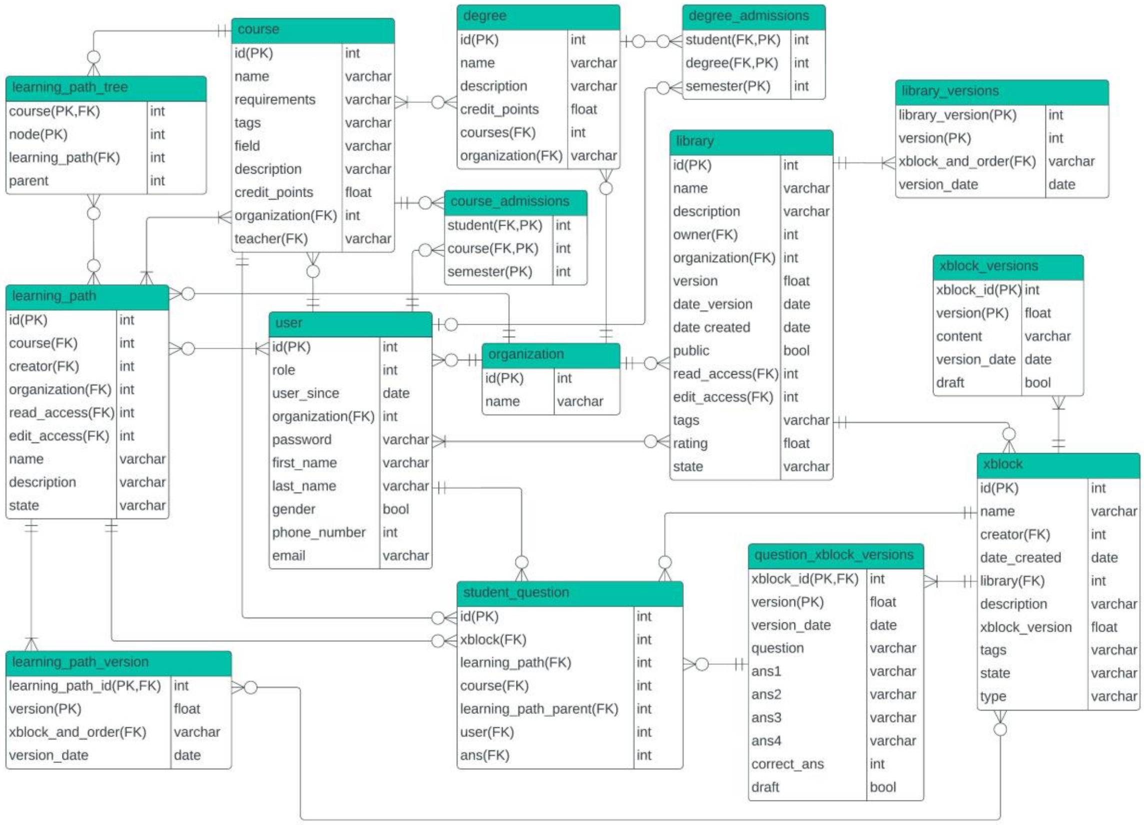


Figure 5: Illustration of how we would structure our database and how we would store everything in our LMS.

Above you can see how we have decided to structure our database. The tables "*user*", "*organization*", "*degree*" and "*course*" are self-explanatory. The "*course_admissions*" and "*degree_admissions*" stores which of the students have access to or are part of which courses and degrees.

The "*xblock*" table stores metainformation about every xBlock. While "*xblock_versions*" store the xBlocks that only needs a single field to store the content. The xBlocks we use that are stored there are text, image, and video. All the tables with *"_version*" in their names keep the older versions of the xBlocks when new versions are made.

Some xBlocks would need to take input from the students. To do this they require their own tables. We demonstrate how we would solve this with our question xBlock. The table "*question_xblock_versions*" store the questions and up to four answers, while "*student_question*" stores all the student's answers and information to keep track of which question the answer belongs to and where the student encountered the question. This is necessary as through sharing the same question could potentially be in multiple learning paths and courses.

The "*library*" table stores metainformation about the libraries while "*library_versions*" keep track of which xBlocks belongs to each library as well as which xBlocks belonged to older versions of libraries.

The "*learning_path*" table stores metainformation about learning paths, while "*learning_path_version*" store which xBlocks the learning path consists of, as well as their order. This also stores older versions.

We have also a table called "*learning_path_tree*" which houses how different learning path trees are structured in different courses.

Some of the entities we store in the database have tables where we store older versions. The reason for this is simple, and that is sharing. When you enable sharing and the creator can change anything they want, you need a way to handle versions. If someone creates a great xBlock and someone else chooses to use it in a learning path, that is great. But if the user that created the xBlock then want to change something about it later, this would automatically change it for the second person and potentially break their learning path. Therefore, we keep older versions of xBlocks, libraries, and learning paths and give the users the option of choosing whether they want to switch to the new version or keep the old.

6.0 Implementing with Craft

We have decided to build a Learning Management System on Craft CMS. The reason for this is Craft's ability to create and manage content, as well as present it, and with Craft's plugin system we hope to be able to build basically anything within our system.

6.1 What is Craft CMS

Craft is a content management system. It was created in 2013 by Pixel & Tonic. It is a WordPress alternative for developers who need deep control and powerful performance from content management tools (Reifman 2015).

Craft is not a site-builder. You will need to build your HTML, CSS, and JavaScript by yourself. Craft is exceptionally scalable, and it has native features for complex content management relationships. It is a good choice for small websites, but Craft will shine with larger content sites that have multi-layered and inter-related content (Reifman 2015).

Craft is written in PHP on the Yii 1.x platform, but you don't need to know PHP or Yii to use Craft CMS (Reifman 2015).

6.1.1 Features of Craft

These are the main features of Craft CMS (CraftCMS(1)).

1. Simple and organized design

- Custom fields: based on your content's specific needs you can set up custom fields with suitable built-in or plugin field types.
- Section types: entries can be managed depending on the use case under three types of sections.
- Relations: you can relate entries, tags, assets, categories, users, and plugin-supplies element types with the relation system.
- Categories and tags.
- Multi-site: the content for multiple sites can be managed from a single Craft installation.
- Localization: your website content can be localized to target specific languages and territories.

- Matrix: multiple content types can be managed in one section and create mixed, repeating types of content.
- Entry types: manage multiple types of content.
- Forms: form builder plugins are available in the plugin store.

2. Easy site management

- Live preview: see how the content will look on the website and share a draft with other people using private preview URLs.
- Asset management: define volumes, subfolders, and view files in list or thumbnail views with Craft's desktop-class asset manager.
- Image editor.
- User-management: manage user groups, set permissions, and allow public user registration.
- Dashboard: each user has their own customizable dashboard. Manage your site from a smartphone or a tablet as the dashboard is fully responsive.
- One-click update.

3. Create and manage an ecommerce store

- Tight integration: manage the products and the website under the control panel.
- Custom order fields: custom fields can be added to orders to track any order information.
- Order management: keep track of in-progress carts and completed orders.
- Products and variants: create product types so you can display similar products to customers.
- Subscriptions: customers can sign up for a subscription, and you can keep track of their subscription status.
- Taxes and shipping: set shipping methods and tax rules.

4. Toolkit

- Plugin store.
- Debug toolbar.
- Templating.

5. Documentation

   Craft CMS comes with an informative set of documentation to guide users to have a better experience with the system.

6. Template system

   Craft's integration with Twig allows for creation of layout-based templates, reusable macros, template logic and function, and template short tags.

In chapter 4 we talked about which features are in a standard LMS and which features we want in our system. These are the features we want that we believe Craft can help solve with its features out of the box:

- Course management – with Craft's ability to manage and design content i.e., with custom fields, section types, entry types etc.
- Adaptability – by being more creative with Craft's field types we can make this possible.
- Shareability – with configurable user permissions.
- Integrations – the plugin store can help integrate third party software.
- Automation – there are plenty of plugins that can help automate anything and everything within the system.
- Surveys – with the forms feature.
- Configurable roles and permissions – with Craft's user management.
- Groups – with user management in Craft.
- Forum – this can be done with user management and content management.
- Analytics – there exists plugins that can showcase website analytics in Craft.
- Learning paths – this can be done with content management, like matrixes, in Craft.
- Notifications & reminders – existing plugins can solve this as well.
- Localization – localization feature in Craft.
- Subscription – a feature in Craft.
- Invoicing – a feature in Craft.
- Live communication – plugin.
- Branding – can host many clients where each client has its own brand.
- Single sign on – plugin.

- Messaging – this can be done with plugins.
- Rating – plugin.
- Security – Craft is trusted by corporations like Microsoft, Apple, and Dell.

Craft solves many of the features we are planning to include in our system. These are some examples:

- The xBlocks can be solved by using field types, as they can be customized as plugins, and can be assigned to parts of the system like sections, assets, users, tags, global sets, and matrix fields.
- Learning paths can be solved in Craft using the matrix field type. This field type is made up of multiple blocks, which again are made up of one or more custom fields.
- Shareability can be done by using Craft's ability to administer user access to certain content, as well as the subscription feature.
- Ratings can be done by implementing plugins that allows for users to rate elements in Craft.
- Craft comes with tools that help with the analytics feature we want.
- There is a plugin for implementing Discord

For the features that Craft does not immediately solve, we will have to extend it with custom plugins and modules.

6.2 Extending Craft

The two best extension features Craft has, are the modules and the plugins. Yii's documentation describes modules like this: "Modules are self-contained software units that consists of models, views, controllers, and other supportive components" (Yiiframework). Which means it extends the system in various ways without the need to change any of the code in the core of the system.

Some of the features we will have to create new plugins or modules for are

Modules can be simple, but they can also be complex. An example of a simple module can be a dashboard widget type, while a complex module could be an entirely new concept to the system, like an ecommerce application. Plugins are a concept from Craft, which means you will not find mentions of it in the Yii documentation. Plugins can do the same things as modules, and some other things that make them better for being publicly released (CraftCMS(2)):

- Plugins can be installed/trialed/purchased from the plugin store in Craft.
- Plugins can make changes in the database when installed, updated, and uninstalled.
- Plugins will get their own settings page in the control panel's settings section.
- Plugins can be enabled or disabled by an admin, without needing to run any Composer commands.

We will need to use modules and plugins to build the features we need in our LMS, that Craft does not offer out of the box.

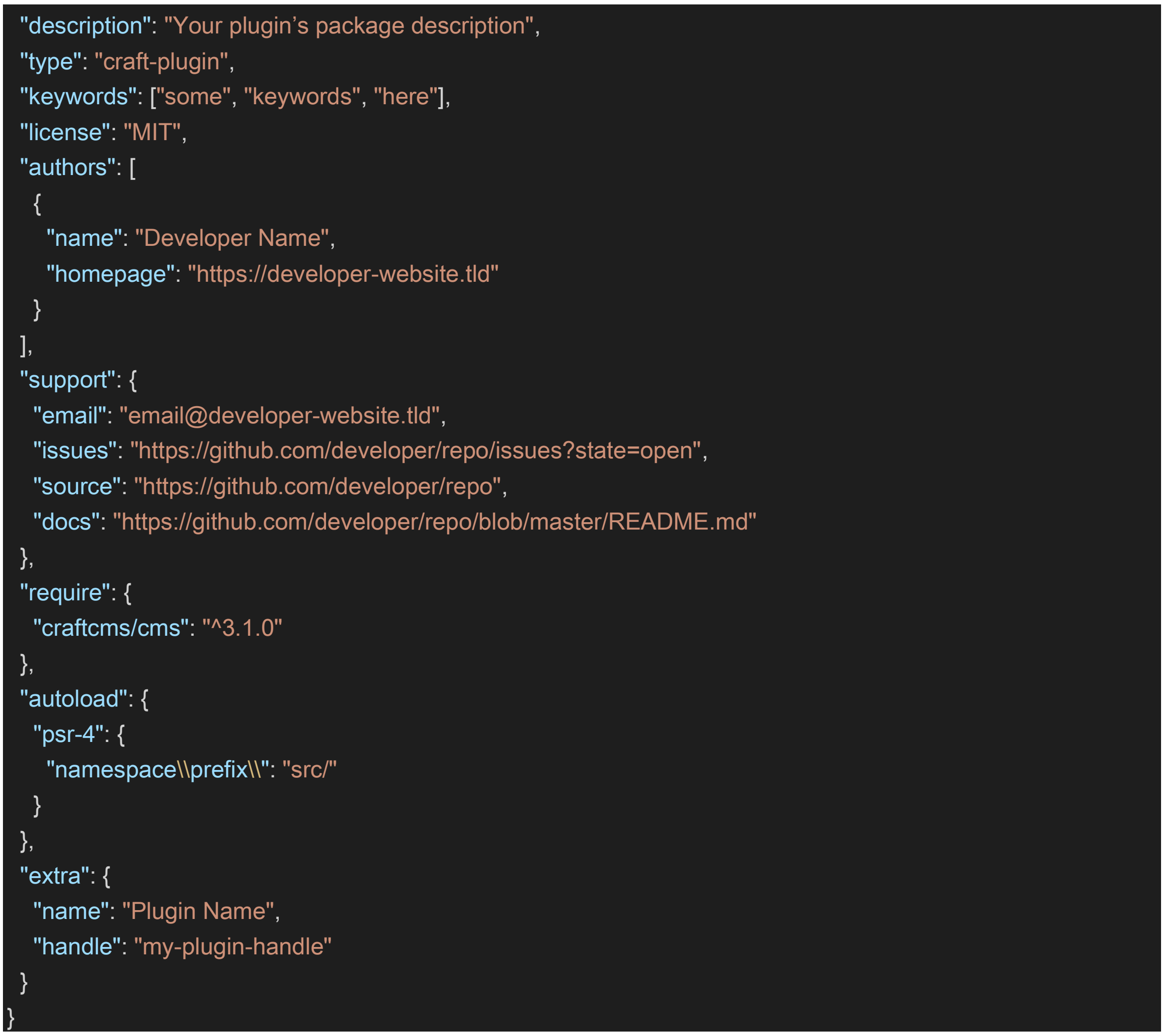
Craft's ability to extend its system with plugins and modules will help when we are trying to create this LMS. xBlocks, which are the cornerstone of the adaptivity in our system, will be different plugins. These plugins we will have to create ourselves, and next section shows how we would do that.

6.3 Plugins

To start working on a plugin, a few things must be decided. The package name for composer, the root namespace, the plugin handle, and the plugin name. The name cannot begin with "Craft" or include any edition-sounding words like "Lite", "Plus", or "Pro". When naming your plugin, choose a name that will age well and save yourself time in the future (CraftCMS(2)).

Then start setting up the basic file structure. Create a directory somewhere on your computer. The name of the directory does not matter. Now create a composer.json file at the root of your directory, and use this as template (CraftCMS(2)):

```
{
  "name": "package/name",
```



```
  "description": "Your plugin’s package description",
  "type": "craft-plugin",
  "keywords": ["some", "keywords", "here"],
  "license": "MIT",
  "authors": [
    {
      "name": "Developer Name",
      "homepage": "https://developer-website.tld"
    }
  ],
  "support": {
    "email": "email@developer-website.tld",
    "issues": "https://github.com/developer/repo/issues?state=open",
    "source": "https://github.com/developer/repo",
    "docs": "https://github.com/developer/repo/blob/master/README.md"
  },
  "require": {
    "craftcms/cms": "^3.1.0"
  },
  "autoload": {
    "psr-4": {
      "namespace\\prefix\\": "src/"
    }
  },
  "extra": {
    "name": "Plugin Name",
    "handle": "my-plugin-handle"
  }
}
```

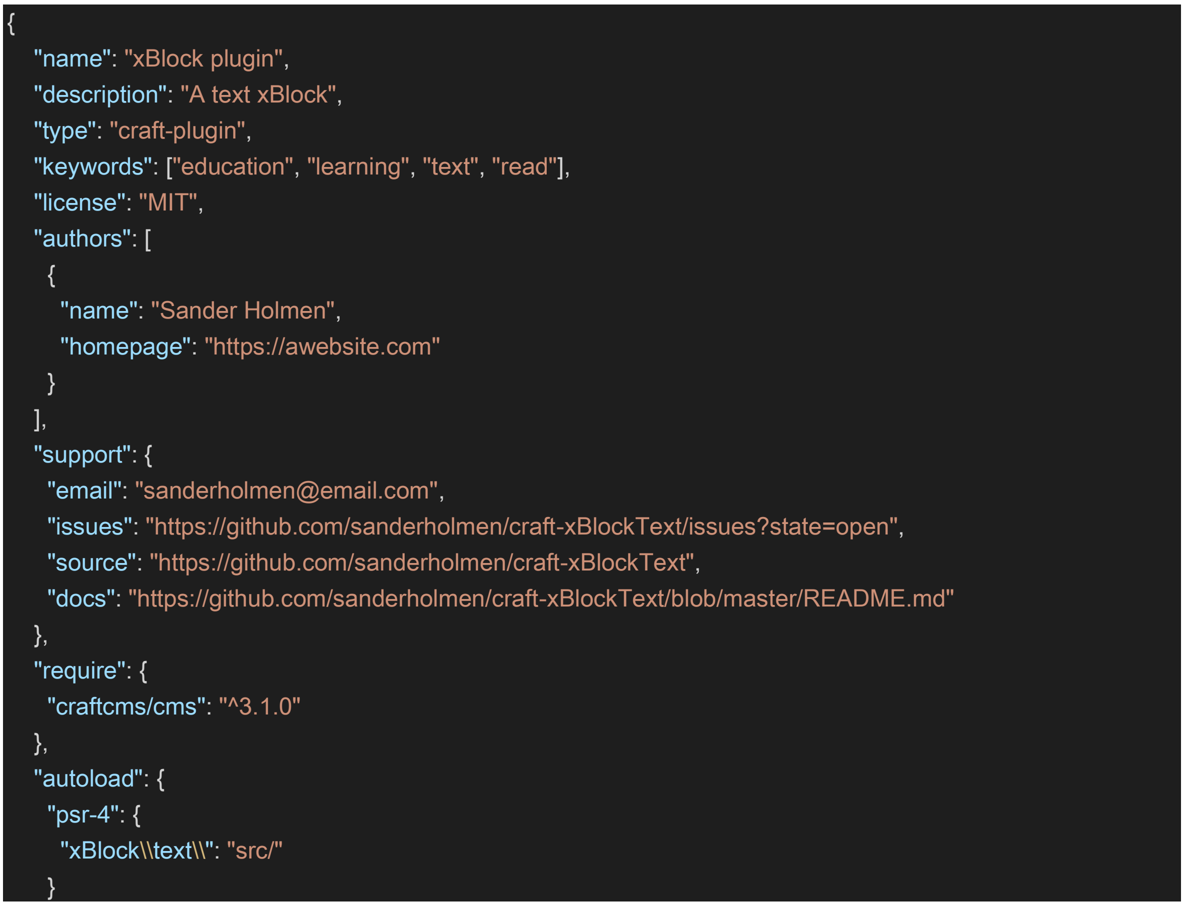
These are the things you need to replace in the script:

- “package/name” with the package name you have decided
- “developer name” with your name the organization that should get the credit
- “https://developer-website.tld” with the website URL
- “email@developer-website.tld” with your support email for the website
- “developer/repo” with the GitHub account and names of the repositories where the plugin will live
- “master” with the primary branch name in your repository
- “namespace\\prefix\\” with the namespace prefix you chose (double backlashes because of json.)
- “Plugin name” with the plugin name you chose
- “my-plugin-handle” with the plugin handle you chose
- “MIT” with “proprietary” if you are planning to use Craft License

In the “extra” object, you can also include a few things:

- class: the plugin class name. If this is not set, at each of the autoload path roots the installer will look for a Plugin.php file
- description: plugin description. If this is not set it will use the main description property
- developer: the developer’s name. If this is not set it will use the “name” property from the “author” object
- developerUrl: the developer URL. If this is not set, the “homepage” property from the “author” object will be used
- developerEmail: the support email. If this is not set it will use the “email” property from the “support” object
- documentationUrl: the URL for the plugin’s documentation. If this is not set it will use the “docs” property from the “support object”

This is what the JSON file could look like when creating an xBlock plugin:



```
{
  "name": "xBlock plugin",
  "description": "A text xBlock",
  "type": "craft-plugin",
  "keywords": ["education", "learning", "text", "read"],
  "license": "MIT",
  "authors": [
    {
      "name": "Sander Holmen",
      "homepage": "https://awebsite.com"
    }
  ],
  "support": {
    "email": "sanderholmen@email.com",
    "issues": "https://github.com/sanderholmen/craft-xBlockText/issues?state=open",
    "source": "https://github.com/sanderholmen/craft-xBlockText",
    "docs": "https://github.com/sanderholmen/craft-xBlockText/blob/master/README.md"
  },
  "require": {
    "craftcms/cms": "^3.1.0"
  },
  "autoload": {
    "psr-4": {
      "xBlock\\text\\": "src/"
    }
  },
  "extra": {
    "name": "xBlock text plugin",
    "handle": "sanders-plugin-handle"
  },
  "repositories": [
      {
        "type": "path",
        "url": "./plugins/*"
      }
    ]
}
```

Now create a src/Plugin.php file. This will be the plugin's entry point for the system. At the beginning of every request this file will get instantiated first (CraftCMS(2)).

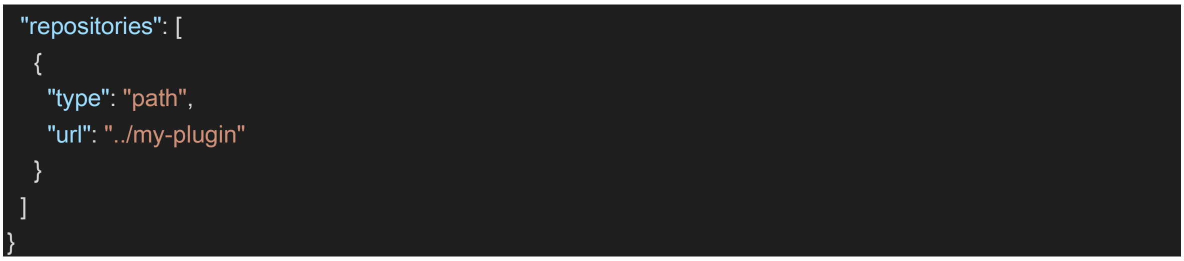
Use this template for the file:

```
<?php
namespace mynamespace;

class Plugin extends \craft\base\Plugin
{
    public function init()
    {
        parent::init();

        // Custom initialization code goes here...
    }
}
```

The init() method is where you should register event listeners and other steps needed to initialize itself (CraftCMS(2)).

Now, to get Craft to see the plugin, it has to be installed as a Composer independency of the Craft project. The best way to do this is with a path repository. This will tell Composer to symlink the plugin into the vendor/ folder. In order to do this, you have to open the composer.json file. Then make the following changes (CraftCMS(2)):

```
{
  "minimum-stability": "dev",
  "prefer-stable": true,
```



```
 "repositories": [
   {
     "type": "path",
     "url": "../my-plugin"
   }
 ]
}
```

Where you replace “../my-plugin” with the path name of your plugin’s root directory.

Now, in your terminal, go to the location of your Craft project. Then tell Composer to require the plugin with this command:

“Composer require package/name”

Where package/name is the name you gave the plugin in the composer.json file.

The package is installed when the log shows this message:

“- Installing package/name (X.Y.Z): Symlink from ../my-plugin”

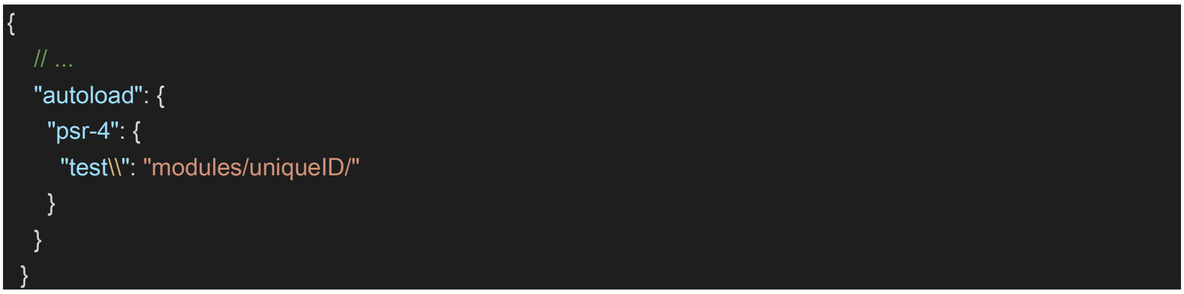
Then, if you want to publicly release the plugin, you can register it as a new Composer package on Packagist, which is a PHP package repository (CraftCMS(2)).

If you want to have an icon for your plugin, which will be visible on the plugin page of Craft, you must save the image as a square SVG file, with the name “icon.svg” at the root of the plugin’s source directory (CraftCMS(2)).

6.4 Modules

Before you can build a new module, you have to decide the namespace, which is the root namespace the module’s classes will live in. It should start with something that identifies you or the project, not with “craft/”. And the module ID, which is something unique that identifies the plugin within your project. It must begin with a letter and only contain lowercase letters, numbers, and dashes (CraftCMS(2)).

Now, when you start building the module, you want to make a directory for it somewhere within your Craft project. An example is: “modules/<ModuleID>/”.
Then you have to tell composer how to find your module’s classes. This is done by creating a composer.json file in your directory and setting the “autoload” field. For example, if the module’s namespace is “test” and it is located at “modules/uniqueID”, this is what it would look like (CraftCMS(2)):



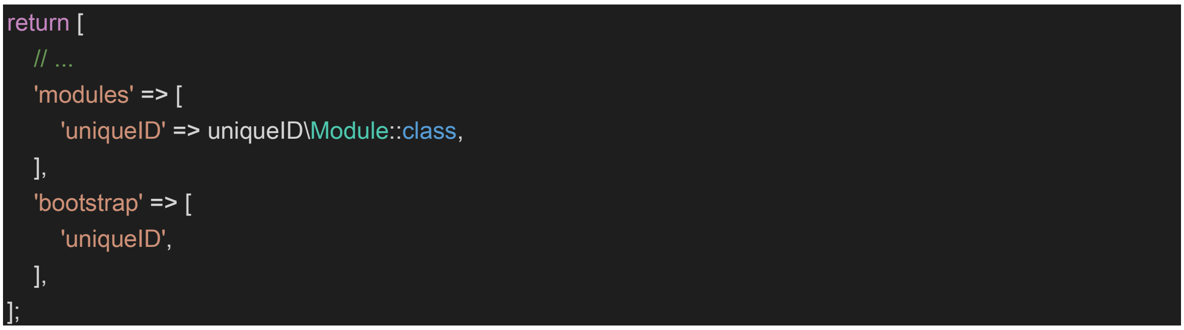
```
{
  // ...
  "autoload": {
    "psr-4": {
      "test\\": "modules/uniqueID/"
    }
  }
 }
```

When that is in place, go to the directory of your project in the terminal and type in the following command:

“composer dump-autoload -a”

Based on the new mapping, this will tell composer to update its class autoloader script.

Add the module to your project’s application configuration which is located at “config/app.php”, by listing it in the “modules” and “bootstrap” arrays. For example, if the module ID is “uniqueID” and its module class name is “uniqueID\Module”, this is what you should add to the file (CraftCMS(2)):

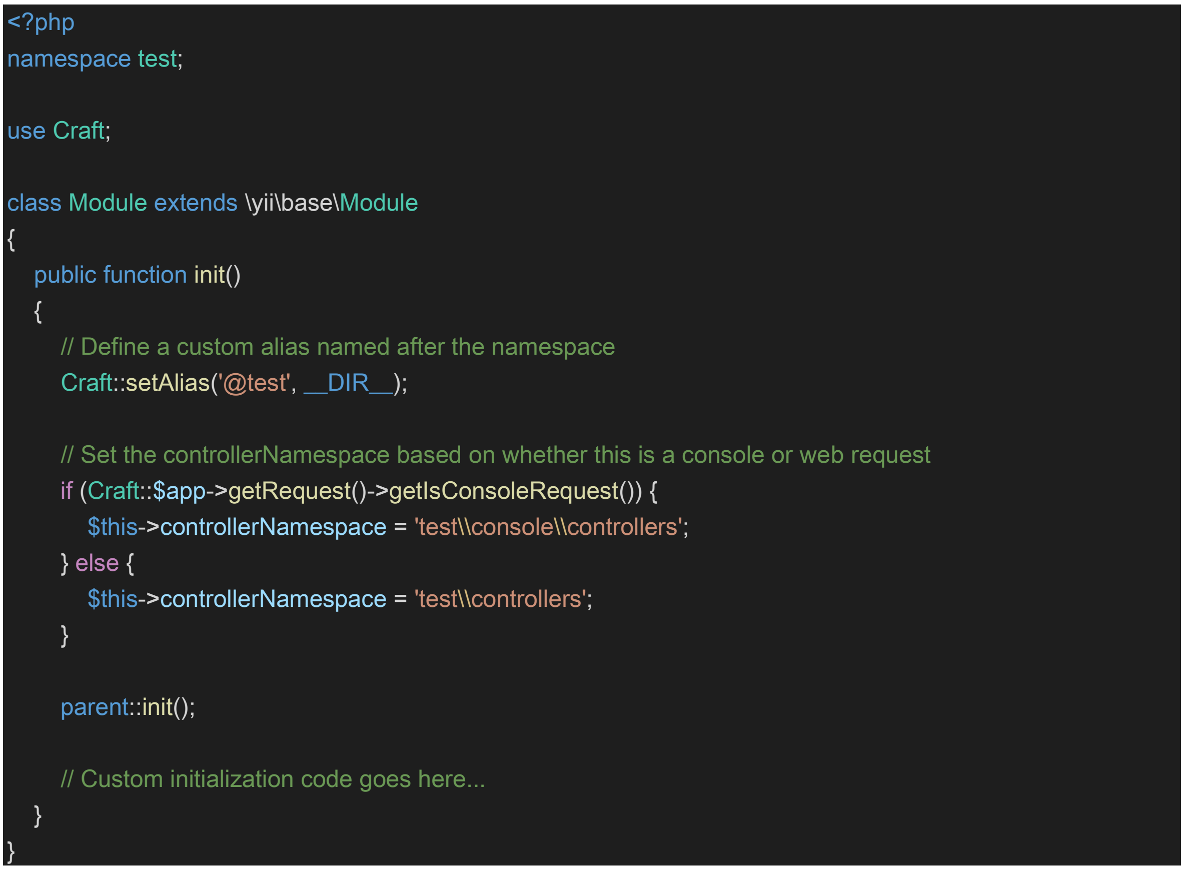


```
return [
  // ...
  'modules' => [
    'uniqueID' => uniqueID\Module::class,
  ],
  'bootstrap' => [
    'uniqueID',
  ],
];
```

You can remove the ID from the “bootstrap” array if the module does not need to get loaded on every request.

The Module.php file you have created in the “modules” folder, is the entry point of the module to the system. The “init()” method is where you should register event listeners and other steps it needs to initialize itself (CraftCMS(2)).

Use this template for your “Modules.php” file:



```php
<?php
namespace test;

use Craft;

class Module extends \yii\base\Module
{
    public function init()
    {
        // Define a custom alias named after the namespace
        Craft::setAlias('@test', __DIR__);

        // Set the controllerNamespace based on whether this is a console or web request
        if (Craft::$app->getRequest()->getIsConsoleRequest()) {
            $this->controllerNamespace = 'test\\console\\controllers';
        } else {
            $this->controllerNamespace = 'test\\controllers';
        }

        parent::init();

        // Custom initialization code goes here...
    }
}
```

Where “test” is the namespace that you chose.

7.0 Business model

Our vision for course material and publishers' books are quite different from the traditional bookstore. Books are expensive to make, and they also have to be converted into xBlocks. And they will then compete with teachers' xBlocks. So how will everyone get paid?

Our plan is largely based on scale and competition. When teachers build their learning paths, every xBlock they use are either free or premium. When students interact with a learning path in a course, they can see every xBlock. The free ones are displayed, and it should not be obvious where one ends and the next begin. Then if they do not have a subscription, they will stumble upon a section grayed out with a padlock on it. Behind the padlock is a premium xBlock they do not have access to.

When student subscribe, they subscribe to the system itself. This gives access to premium xBlocks in every course they take. The money from the subscription is then distributed between the teachers and publishers based on which premium xBlocks the students have interacted with. The students can also choose to buy a discounted book in which case the money from that purchase goes to the publisher.

This income is based heavily on the numbers of students that use the system. In the beginning it will not be the best, but as time goes on and hopefully the systems market share increases there will be enough money to attract publishers. A lot of textbooks are published by teachers and those who are passionate about writing books might also be tempted as individuals to create xBlocks and libraries for our system.

8.0 Conclusion

With the design decisions we have made, we believe this system will be a good LMS suited for hybrid learning. Craft helps with a strong well-tested CMS with many features. The features that Craft does not cover we have a solution for with Craft’s ability to be extended with custom plugins and modules.

By splitting course material into what we call xBlocks, we believe we have found a way to make course-creating easier than it is today. It also makes adaptive learning possible with learning paths created from these xBlocks.

Our solution with learning paths as course structure opens a wide range of possibilities within a course. Teachers can now include more types of students in the same course. This means the locations, programs, or prior knowledge of the students matter less, as the different learning paths can be tailored for different types of students.

Discord as a plugin in our system will make for the ability to stream classroom lectures to online students, as well as it being a great communication platform among students and teachers.

As well as making hybrid teaching easier, we are making it possible to share course material in an easy way. This means course material can be borrowed from teachers within the same institution. We also make it possible for teachers in different institutions to borrow material from each other. The creator of the content decides if there is a fee for using it, or if it is free to use.

The bookstore is a new concept we are introducing. This creates an opportunity for publishers to participate and continue to earn a fair share as the development of modern education results in physical books being used less.

The business model we have chosen opens a possibility for teachers to earn money from creating course material. This creates an incentive for teachers to make good course material and gives an opportunity for people to create course material full time as a job.

9.0 Future Work

In this report we have talked about the design and how we would develop an LMS suited for hybrid learning. We have not done any actual programming to develop the system. We believed we would, but designing it was harder and took more work than we anticipated. The next steps we would take if we wanted to continue working on this system and bring it to life would be the following.

- Continue to study Craft CMS and really get to know it. By getting to know it as well as possible, we could potentially save us a lot of work and our mistake count would probably be lower.
- Decide which Craft field types we would use for each type of xBlock. Work out which existing plugins we would use for certain tasks.
- As Craft is just a content management system without a front end we would have to design and code how the user interfaces to look. This includes sites for the public, like homepages, as well as the everything the students interact with as they attend courses.
- A lot of the features we would like for our system to have can be attained either through built in features in craft, or through existing plugins. But for some we would have to create our own plugins. This would be extending craft as we explained earlier in our report.
- There would obviously have to be a lot of testing, but eventually we would need to cooperate with an institution like Molde University College to do a demo course system test with students.

References

Bates, A.W. 2019. "Teaching in a Digital Age – Second Edition". Pressbooks. https://pressbooks.bccampus.ca/teachinginadigitalagev2/

Bradley, V. M. 2021. “Learning Management System (LMS) use with online instruction.” *International Journal of Technology in Education (IJTE)* 4(1): 68-92. https://doi.org/10.46328/ijte.36

CraftCMS(1). https://craftcms.com/features

CraftCMS(2). https://craftcms.com/docs/#browse-product-documentation

Hornshaw, Phil. 2021. “What is Discord?” Digitaltrends. https://www.digitaltrends.com/gaming/what-is-discord/

Jung, Sooyoung and Jun-Ho Huh. 2019. “An Efficient LMS Platform and Its Test Bed” *Electronic* 8(2): 154. https://doi.org/10.3390/electronics8020154

Kunju, Sageena P. 2021. “What are the different types of LMS out there.” Edly. https://edly.io/blog/what-are-the-different-types-of-lms-out-there/

Lynn, Katie-Anna. 2021. “Top 29 learning management systems features you should consider.” LearnUpon. https://www.learnupon.com/blog/lms-features-overview/

Mirata, Victoria, Franziska Hirt, Per Bergamin and Christo van der Westhuizen. 2020. “Challenges and contexts in establishing adaptive learning in higher education: findings from Delphi study.” *Int J Educ Technol High Educ* 17, 32. https://doi.org/10.1186/s41239-020-00209-y

Morze, N, L Varchenko-Trotsenko, T Terletska and E Smyrnova-Trybulska. 2021. “Implementation of adaptive learning at higher institutions by means of Moodle LMS.” J. Phys.: Conf. Ser. 1840 012062

Nielson, Bryant. 2015. “What’s the difference between a MOOC and an LMS?” Your training edge. https://www.yourtrainingedge.com/whats-the-difference-between-a-mooc-and-an-lms/

Normadhi, Nur Baiti Afini, Liyana Shuib, Hairul Nizam Md Nasir, Andrew Bimba, Norisma Idris and Vimala Balakrishnan. 2019. “Identification of personal traits in adaptive learning environment: Systematic literature review.” *Computers & Education* Vol 130. 168-190. https://doi.org/10.1016/j.compedu.2018.11.005

Polakowski, Marek. 2021. “Open source LMS comparison | Moodle vs OpenEdx vs Canvas LMS.” Selleo. https://selleo.com/blog/open-source-lms-comparison

Popupsmart. 2022. “What is Craft CMS?” https://popupsmart.com/encyclopedia/craft-cms

Ramani, Ramesh. 2017. “2017 Preview: 11 Learning Management System Technology Trends by the Industry’s Top Analysts.” eLearning Industry.

https://elearningindustry.com/11-learning-management-system-technology-trends-top-analysts-2017-preview

Regjeringen. 2021. «Strategi for digital omstilling i universitets- og høyskolesektoren.» Updated 10. September, 2021. https://www.regjeringen.no/no/dokumenter/strategi-for-digital-omstilling-i-universitets-og-hoyskolesektoren/id2870981/

Regjeringen. 2017. «Digitaliseringsstrategi for universitets- og høyskolesektoren» Updated 19. September, 2017. https://www.regjeringen.no/no/dokumenter/digitaliseringsstrategi-for-universitets--og-hoyskolesektoren---/id2571085/

Reifmann, Jeff. 2015. "Introduction to Craft CMS." Tutsplus. https://code.tutsplus.com/tutorials/introduction-to-craft-cms--cms-22982

Salah, Saeed, Mahmoud Thabet. 2021. «E-learning management systems: A feature based comparative analysis». *JISTEM J.Inf.Syst. Technol. Manag.* 18. https://doi.org/10.4301/S1807-1775202118003

Scerbakov, Alexei, F, Kappe, N, Scerbakov. 2019. «Security vulnerabilities in modern LMS." Researchgate. https://www.researchgate.net/profile/Nick-Scerbakov/publication/334524100_SECURITY_VULNERABILITIES_IN_MODERN_LMS/links/5d2f8df2458515c11c392fda/SECURITY-VULNERABILITIES-IN-MODERN-LMS.pdf

Serhat, Kurt. 2021. "Adaptive learning: What is it, what are its benefits and how does it work?" Educational Technology. https://educationaltechnology.net/adaptive-learning-what-is-it-what-are-its-benefits-and-how-does-it-work/

Reifman, Jeff. 2015. "Introduction to Craft CMS." https://code.tutsplus.com/tutorials/introduction-to-craft-cms--cms-22982

Yiiframework. "The definitive guide to Yii 2.0." https://www.yiiframework.com/doc/guide/2.0/en/structure-modules

Acronyms

LMS – learning management system

CMS – content management system

HTML – hypertext markup language

CSS – cascading style sheets

URL – uniform resource locator

JSON – JavaScript object notation

UI – user interface