

# Study, design, implementation and evaluation of a Learning as a Service (LaaS) platform in the cloud



Technological Educational Institute  
of Crete

Department of Informatics Engineering

by

**Giannis Halvatzakis**

20 November 2018



# Abstract

In recent years, the use of online interactive learning platforms has grown to a greater extent, both to strengthen the traditional way of teaching and to fully teach one or more courses.

Harnessing cloud computing infrastructure for the installation and operation of such platforms is a necessary condition to support a large number of users at all times and regardless of location learners (eg Massive Open Online Courses-MOOC). It should be noted that free access to such platforms contributes to the "democratization" of knowledge as well as to the promotion and support of lifelong learning. However a number of issues such as assessment / certification of the performance of a user and the user feedback data that will help in the learning process, remain open.

In this context, this thesis, using teaching material from the Wireless Networks course, will focus on exploring such issues by designing, implementing and evaluating a Learning as a Service (LaaS) platform that will be provided over cloud computing infrastructure.

Giannis Halvatzakis  
November, 2018



# Σύνοψη

Τα τελευταία χρόνια έχει αυξηθεί σε μεγάλο βαθμό η χρησιμοποίηση δικτυακών διαδραστικών πλατφορμών εκπαίδευσης τόσο για την ενίσχυση του παραδοσιακού τρόπου διδασκαλίας όσο και για την εξολοκλήρου διδασκαλία ενός ή περισσότερων μαθημάτων. Η αξιοποίηση υποδομών υπολογιστικού νέφους, για την εγκατάσταση και λειτουργία τέτοιου είδους πλατφορμών είναι μια απαραίτητη προϋπόθεση, για την υποστήριξη ενός μεγάλου αριθμού χρηστών όλες τις ώρες και ανεξαρτήτου τοποθεσίας των εκπαιδευόμενων (π.χ. Massive Open Online Courses - MOOC ).

Πρέπει να επισημανθεί ότι η ελεύθερη πρόσβαση σε τέτοιου είδους πλατφόρμες συμβάλει στη 'δημοκρατικοποίηση' της γνώσης, καθώς και στην προώθηση και υποστήριξη της δια βίου εκπαίδευσης. Παρόλα αυτά μια σειρά από ζητήματα, όπως η αξιολόγηση/πιστοποίηση της επίδοσης ενός χρήστη και η ανατροφοδότηση του χρήστη με δεδομένα που θα τον βοηθήσουν στη μαθησιακή διαδικασία, παραμένουν ανοιχτά.

Στα πλαίσια αυτά η παρούσα πτυχιακή, χρησιμοποιώντας εκπαιδευτικό υλικό από το μάθημα "Ασύρματα Δίκτυα", θα επικεντρωθεί στην διερεύνηση τέτοιου είδους θεμάτων σχεδιάζοντας, υλοποιώντας και αξιολογώντας μια Learning as a Service (LaaS) πλατφόρμα ή οποία θα παρέχεται πάνω από υποδομή υπολογιστικού νέφους.



# Acknowledgements

This bachelor thesis is my first academic milestone.

Firstly I would like to thank my thesis supervisor Evangelos Pallis, for providing much-needed advice and patient tutelage, during my thesis.

Moreover i would like to express my gratitude to Anargyros Sideris for the guidance and his advices.

Furthermore i would like to thank my friends from Heraklion especially Spyros Chiotakis and Fotis Koutoulakis or their support and their help becoming passionate about my work.

Most of all, I would like to thank my family, who always supported me throughout my life.





# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Background . . . . .	13
1.2	Thesis Structure . . . . .	13
<b>2</b>	<b>Theory</b>	<b>15</b>
2.1	Learning . . . . .	15
2.1.1	Traditional Learning . . . . .	15
2.1.2	E-Learning . . . . .	15
2.1.3	The 7 principles of the new learning[8] . . . . .	17
2.1.4	Computerized adaptive testing . . . . .	19
2.2	Cloud Computing . . . . .	20
2.2.1	Key Features . . . . .	20
2.2.2	Service Models . . . . .	21
2.2.3	Deployment Models . . . . .	21
2.3	Web Applications . . . . .	23
2.3.1	Comparison between Web applications and Native applications . . . . .	23
2.3.2	Model–view–controller (MVC) Architecture . . . . .	24
2.3.3	Web application framework . . . . .	25
2.3.4	Dynamic Adaptive Streaming over HTTP . . . . .	25
2.3.5	ABR: Adaptive Bitrate Streaming . . . . .	25
2.3.6	BOLA: Near-Optimal Bitrate Adaptation for Online Videos . . . . .	25
<b>3</b>	<b>Implementation</b>	<b>27</b>
3.1	FrontEnd . . . . .	27
3.1.1	Material Design . . . . .	27
3.1.2	Javascript . . . . .	27
3.1.3	Service Workers . . . . .	28
3.2	Backend . . . . .	29
3.2.1	Apache . . . . .	29
3.2.2	PHP . . . . .	29

3.2.3	Storage . . . . .	30
3.2.4	Database . . . . .	30
3.2.5	Video Streaming . . . . .	30
3.2.6	Virtualization . . . . .	31
<b>4</b>	<b>Results</b>	<b>33</b>
4.1	Authentication . . . . .	33
4.1.1	Auth Implementation . . . . .	33
4.1.2	Login . . . . .	33
4.1.3	Register . . . . .	33
4.2	Course Creation . . . . .	36
4.2.1	Course Details . . . . .	36
4.2.2	Lesson Builder . . . . .	36
4.2.3	Quiz Builder . . . . .	37
4.3	Course Browser . . . . .	41
4.4	Course Enrollment . . . . .	41
4.4.1	Course Dashboard . . . . .	42
4.4.2	Course Quizzes . . . . .	45
4.5	Experience System . . . . .	48
<b>5</b>	<b>Conclusions and Future work</b>	<b>49</b>
5.1	Conclusions . . . . .	49
5.2	Future work . . . . .	49
	<b>Bibliography</b>	<b>51</b>





# Chapter 1

## Introduction

This chapter describes the knowledge background to accomplish this thesis. Also, motivation and related work for the completion of this thesis is discussed. Lastly, follows a brief overview of the thesis structure.

### 1.1 Background

Technological progress has made huge impact in education, changing the whole learning process in many ways.

However a number of issues such as assessment / certification of the performance of a user and the user feedback data that will help in the learning process, remain open.

This thesis implements a e-learning platform in the cloud and uses some techniques for improving the e-learning process.

### 1.2 Thesis Structure

**Chapter 1** contains general information about E-learning and the features which were implemented. Then follows discussion about the motivation and related work about this thesis.

**Chapter 2** contains theory about E-Learning, types of E-Learning, the new learning[8] system, Cloud Computing and theory about modern software engineering.

**Chapter 3** includes implementation details for the E-Learning Platform, the technologies and the tools that will be used.

**Chapter 4** contains the results of the E-Learning Platform.

**Chapter 5** contains the conclusions about the bachelor thesis and some thoughts about future optimizations and features.

# Chapter 2

## Theory

This chapter contains theory about Learning, Cloud Computing and Web Applications. It also includes information about Web application frameworks and MPEG-DASH.

### 2.1 Learning

In a broad sense education includes all activities designed to influence specifically the thought, character and physical education of the person. From a technical point of view, education is the process by which society deliberately transmits knowledge, skills and values accumulated from one generation to another.

#### 2.1.1 Traditional Learning

#### 2.1.2 E-Learning

E-Learning is the process in which one learns - trained through the use of computers.

Training with this process can be divided into online and offline education. E-Learning online is the viewing of educational material over the Internet (from a website) while offline learning is the display of educational material stored on our computer, educational cdroms etc.

E-Learning is defined as the use of electronic media, educational technologies as well as information and communication technologies (ICT) in education. Electronic Learning uses different types of electronic media (such as video, audio, satellite, CD-ROM, flash drives) and technologies (text, image, video & audio, animation) and assisted or computer-assisted or computer-based learning, on local networks (intranet / extranet) or web-based learning. For this reason, e-learning is not limited to acquiring digital literacy (ie acquiring ICT skills) but may include different types and methodologies, such as software, Internet,

CD-ROM or any other form of electronic or interactive media. E-Learning can take place both inside and outside the classroom. It can be either self-directed or instructor-led. It can also be asynchronous or synchronous. Electronic Learning can be used as a tool in distance education and training as well as supportive in conventional face-to-face teaching. The term used for this purpose is mixed or hybrid training.

In literature, many terms are often synonymous with e-learning. Some of them are multimedia education, computer-based instruction, online education / web-based education, digital education, mobile / portable training ( m-learning) and others. Each of these terms has been devised to describe or emphasize a particular educational technology, approach or feature, but all are forms of e-learning. For example, the definition of mobile learning is used to train through mobile devices, emphasizing the concept of learning mobility. However, there are no other distinct differences of the term in relation to e-learning.

Notwithstanding the above, a distinction with more defined boundaries, but which would be useful to highlight, is that between e-learning, distance learning and open education. Distance learning requires the concept of distance (that the trainer and the trainee are in a different place) without necessarily requiring the use of an electronic medium (eg, sending educational material by mail), while open [ refers to a much more flexible way of training than the rate of study, length of study completion, and admission criteria (grades, ages, etc.) [3].

## **Types of E-Learning**

E-learning is divided into Synchronous and Asynchronous Learning[4].

**Synchronous learning** is used to describe types of learning and teaching that take place at the same time, but not in the same space. Synchronous learning requires the simultaneous participation of the trainer and trainee, where the interaction between them is done in real time and the participants can exchange opinions as well as educational material. Web conferencing, video conferencing, chatting, VoIP telephony, live streaming live broadcasting, online seminars (webinars) and more. all are forms of synchronous learning[5].

**Asynchronous learning** is used to describe forms of learning and teaching that take place both in different locations and at different times . Asynchronous Education does not require the simultaneous participation of the trainer and trainee, and the participants can choose their own personal time frame and collect the training material accordingly. E-mail, video conferencing, podcasts, forums, etc. are forms of asynchronous learning.



Asynchronous education is more flexible than modern. This kind of education is self-teaching, semi-autonomous education and co-operative education.

### 2.1.3 The 7 principles of the new learning[8]

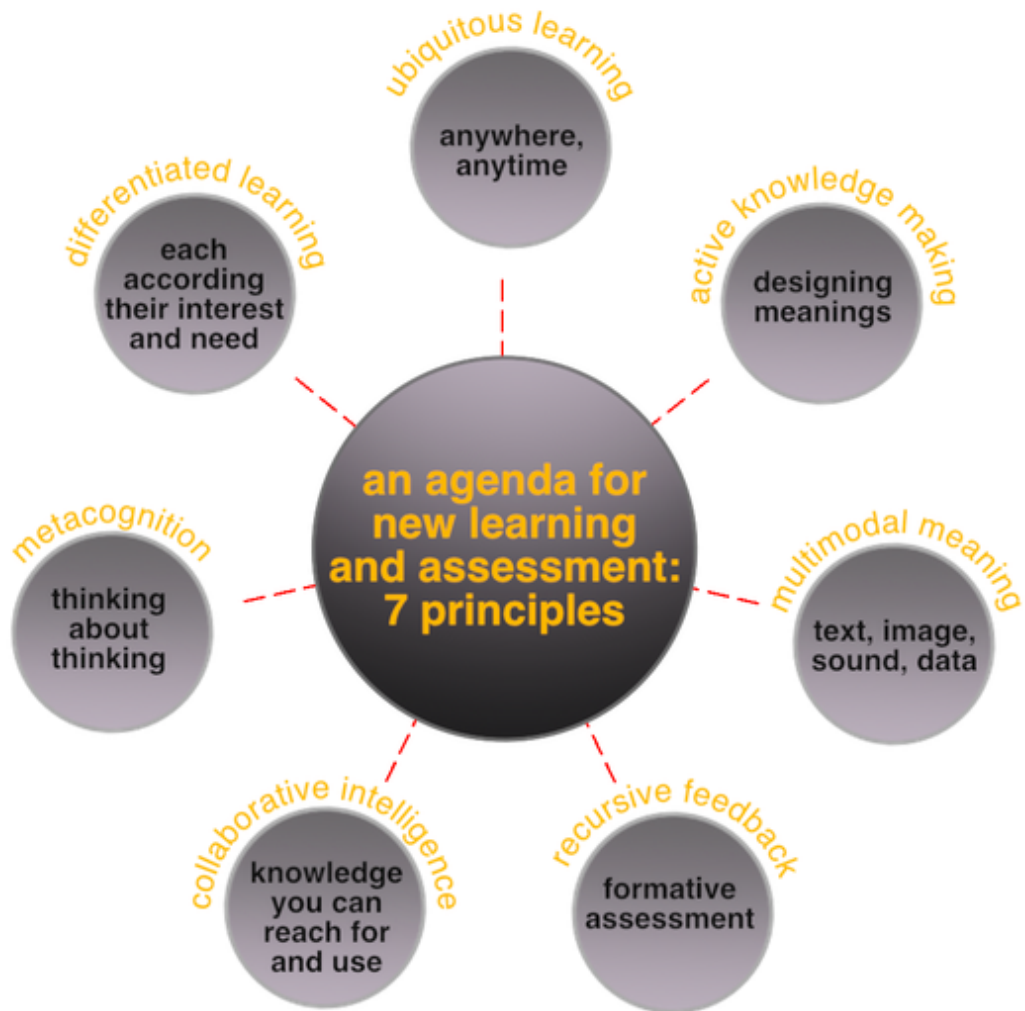


Figure 2.1: The 7 principles of learning

#### Ubiquitous learning

Students are able to learn anywhere, anytime. Ubiquitous learning is closely related with mobile devices and the learning environments can be accessed in various contexts and situations.

## **Active knowledge making**

Students become producers of active knowledge, less than they are consumers of knowledge. Active knowledge-making practices support the contemporary emphasis on innovation, creativity and problem solving - attributes of "knowledge economy" and "knowledge society" par excellence.

## **Multimodal meaning**

Today's students should be able to use digital media to compare and link a variety of media types such as text, video, audio, diagrams and other types of presentable information. In all the thematic areas, the meaning of representations of creation and knowledge is supported and improved today by digital production skills and technologies.

## **Recursive feedback**

A new generation of assessment systems, including continuous human assessment mediated by a machine from multiple prospects (including peers from multiple sources) and machine feedback (e.g. using natural language processing). Using Data mining techniques can also assist in the evaluation of the student work. The student is also offered immediate feedback, or an evaluation that is for learning (formative assessment) and not just of learning (summative assessment).

## **Collaborative intelligence**

peers offer each other structured feedback, knowledge resources available are diverse and open, and the contributions of peers and sources to knowledge formation are transparent and documented.

This builds the transversal collaboration and negotiation skills necessary for a complex and diversified world. It focuses on learning as a social activity rather than learning as an individual memory.

## **Metacognition**

Involve the feedback offer and feedback and recruit the students as independent evaluators and peers.

It is used to describe the collection of information about what the person feels and realizes about the knowledge, beliefs, thoughts, judgments and feelings that characterize it at a specific time. These considerations do not coincide with the processes that take place at the time of the cognitive process or the solution of a problem, but relate to what the individual believes in his knowledge in relation to various cognitive subjects.

## **Differentiated instruction**

Students need not be on the same page, nor in the same chapter, nor do they do the same task. The new way of learning gives flexibility so as to the student evolves according to his / her needs and ways of learning whom he wants. Students do not need to work on the same material at the same time as it gives diversity in the classroom.

### **2.1.4 Computerized adaptive testing**

Computerized adaptive testing is a modern assessment method which allows the teacher to create tests that efficiently measure the taker's abilities. Adaptive testing questions are saved in a question bank and tagged with a difficulty score. The questions are chosen based on the ability level of the test taker. If the test taker answers correctly on a question, a more difficult question is presented next. If the test taker answers a question falsely, a easier question is presented next. At the end of the test, the users ability will be estimated based on the user's choices. The test stops when the system is confident about the user's difficulty level[10].

## 2.2 Cloud Computing

According to NIST [9] Cloud computing is Network-based computing in which large groups of remote servers are connected to enable sharing of data processing activities, centralized data storage, and online access to IT services or resources. It is a model that allows flexible, on-demand network access to a shared set of customizable computing resources (such as networks, servers, storage, applications and services) which can be quickly delivered and delivered with minimal effort to manage or interact with the service provider. This cloud model promotes availability and consists of **five** key features, **three** service models, and **four** development models.

### 2.2.1 Key Features

#### **On-demand self-service**

A user can provide calculation capabilities such as storage and server time, according to the needs automatically without requiring human interaction with each service provider.

#### **Broad network access**

The services are available over the network and can be accessed from multiple client platforms such as desktops, laptops and mobile phones.

#### **Resource pooling**

The computing resources of the provider are grouped to serve more consumers using a multi-tenant model, with different physical and virtual resources that are dynamically allocated and reallocated according to consumer need.

#### **Rapid elasticity**

The available provision capabilities should appear to be unlimited for the customer and may be available in any quantity at any time. The capabilities can be provided and released elastically, in some cases automatically, in order to scale according to demand.

#### **Measured service**

Cloud systems automatically monitor and optimize the use of resources by taking advantage of a measurement capability at a level of abstraction appropriate to the type of service. The use of resources can be monitored, controlled and reported, guaranteeing transparency both for the provider and for the consumer of the service used.

## **2.2.2 Service Models**

### **Software as a Service**

Software-as-a-Service (SaaS) is a software license model in which software access is provided by subscription, with software located on external servers rather than on servers located internally. Typically, software as a service is accessible via a web browser, with users accessing the system using authentication methods such as email and password. Instead of having to install the software on computer, the user is able to access the program via the Internet. Companies commonly use SaaS to manage customer retention, human resources and procurement.

### **Platform as a Service**

Platform as a service (PaaS) is a service model in which a third-party vendor delivers hardware and software tools, usually those needed for developing applications to users on the Internet. A PaaS provider hosts software and hardware. As a result, PaaS frees users from having to install internal hardware and software to develop or run a new application.

### **Infrastructure as a Service**

Infrastructure as a service (IaaS) is a service model that provides outsourced computer infrastructure to support business operations. Typically, IaaS provides hardware, storage, servers and space for data centers or network components.

## **2.2.3 Deployment Models**

### **Private Cloud**

Private cloud refers to a cloud computing model in which IT services are provided on a private IT infrastructure for the dedicated use of a single organization. A private cloud is usually managed through internal resources.

### **Public Cloud**

Cloud infrastructure is made available to the general public or a large industrial group and is owned by an organization that sells cloud services.

### **Community Cloud [2]**

The community cloud in computing is a collaborative effort in which the infrastructure is shared between different organizations in a specific community with common concerns, managed internally or by third parties and hosted internally or externally. It is controlled

and used by a group of organizations that have common interests. Costs are spread across a smaller number of users than a public cloud (but more than a private cloud), so only some of the potential costs of cloud computing are realized.

## **Hybrid Cloud**

With hybrid cloud, a user can connect and share infrastructure and application resources between the cloud and existing resources that are not in the cloud. Using hybrid cloud, an organization is able to extend its existing resources which are stored locally with cloud resources.

## 2.3 Web Applications

In computer science, a web application is a client-server computer program where the client runs it through a web browser.

Most Web applications rely on the client-server architecture in which the client enters information while the server stores and retrieves information. E-mail is an example of this, with most IT companies offering web-based email clients.

Most Web applications rely on the client-server architecture in which the client enters information while the server stores and retrieves information. E-mail is an example of this, with companies such as Google's Gmail and Microsoft Outlook offering web-based email clients.

Currently, web applications have been evolved to the point that can provide offline access and store information locally. An example of this, is an implementation of a word document that stores documents locally and does not require a server for storing data.

Web applications have also the benefit of working across multiple platforms.

### 2.3.1 Comparison between Web applications and Native applications

In the previous computing models such as client-server, the processing load of the application was shared between code installed on each client locally and code on the server side. An application had its own client program that functioned as a user interface and had to be installed separately on each user's personal computer. An update to the application server-side code would typically also require a client-side code update installed on each user's computer, adding less support and significant productivity costs. Furthermore, both the client and application server components were typically tightly linked to a particular architecture and operating system, and their porting to others was often prohibitive for all but the largest applications. (Nowadays, native mobile apps are also blocked by some or all previous issues).

Web applications which are using scripting languages such as HTML, CSS, Js and WebAssembly are supported by a variety of web browsers. Web applications can be described as client - server software in which the client is downloading the software on his computer when he visits a web page, using standard application protocols such as HTTP. Software updates on the client side can occur every time the Web page is visited. During the session, the Web browser is acting as a universal client for any Web application.

Native computer applications remain to this day the most popular applications and this is due to the fact that native applications were the first applications were created.

### 2.3.2 Model–view–controller (MVC) Architecture

Model-View-Controller or MVC is a software architectural pattern for GUIs that was launched in 1978 and it's very popular for web application development.[1] The pattern is composed of three types of modules with three different responsibilities: models, views and controllers.

- A model contains the data to display.
- A view contains the presentation of the graphical interface.
- A controller contains the logic concerning the actions performed by the user.

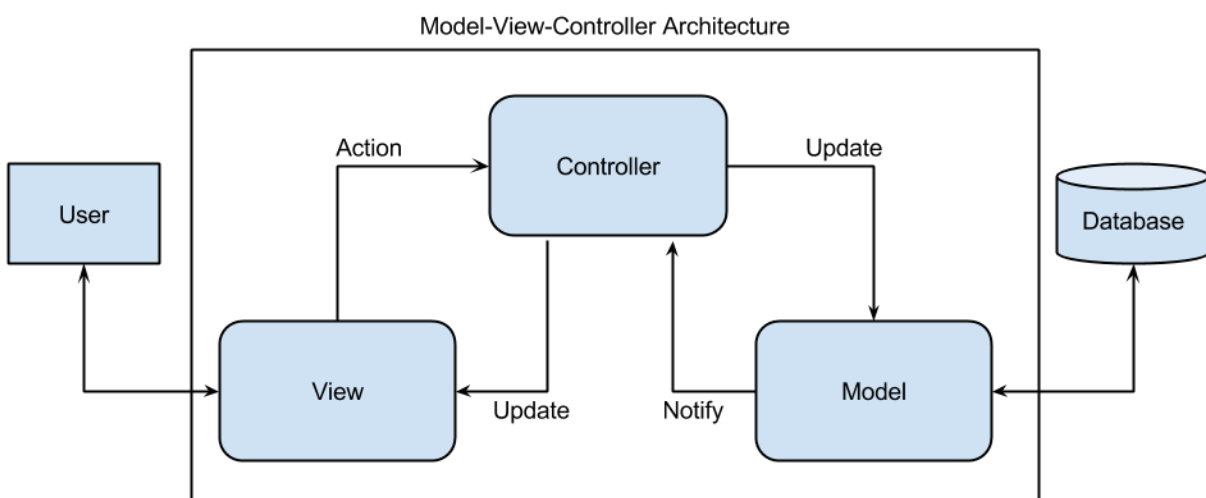


Figure 2.2: MVC Architecture Overview

#### Model

Model is a set of functions (e.g. CRUD functions) that are responsible for managing database data. The model updates the corresponding representation views and controllers when there are changes in the data.

#### View

The visual representation of a model is called as View. It's the result that the framework provides to the browser in order to render the data. (commonly into HTML code). For example, an ecommerce website will display a list of products on a catalog screen.



## Controller

The Controller creates a link between the model and the view. The controller is able to process input, communicate with the model and choose which action should be performed. For example, in a blog instance, the controller will look for the last comments for a blog post and switch them to the render view.

### 2.3.3 Web application framework

Web application frameworks are software tools that simplify the writing, maintenance, and scalability of Web applications. They provide libraries that simplify usual Web development tasks, including routing URLs to appropriate handlers, database interaction, support sessions and user authorization, formatting output such as JSON and improved security.

### 2.3.4 Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH or MPEG-DASH), is an adaptive bitrate streaming technique that enables the streaming of multimedia content over the Internet provided by standard HTTP Web servers.

MPEG-DASH uses existing HTTP web server infrastructure. It allows a variant of devices such as mobile phones and smart TVs to stream multimedia content delivered from the cloud, using multiple Internet receiving conditions.

### 2.3.5 ABR: Adaptive Bitrate Streaming

Adaptive bitrate streaming (ABR) [7] is a technique used in streaming multimedia over the internet. ABR takes into consideration device parameters such as the internet connection status, CPU loads and display resolution in order to stream the multimedia content with the best possible quality.

### 2.3.6 BOLA: Near-Optimal Bitrate Adaptation for Online Videos

BOLA[6] is a video bitrate adaptation and prediction mechanism for HTTP adaptive streaming.

It's an ABR algorithm that determines bitrate from buffer occupancy. As the buffer occupancy grows, higher-bitrate segments are preferred. As the buffer occupancy drops, lower-bitrate segments are preferred.

BOLA takes into consideration the estimated bandwidth of the available network and the expected buffer occupancy level in order to respond proactively and intelligently to current conditions. This leads to two contributions: first, it allows HAS players to take

appropriate actions, before existing methods, to prevent playback interruptions caused by the buffer underrun, reducing the ON-OFF traffic phenomena associated with current approaches and increasing the QoE. Secondly, it facilitates the equal sharing of bandwidth between players competing with the bottleneck connection.

# Chapter 3

## Implementation

This chapter contains details about the technologies that will be used for developing the LaaS application.

### 3.1 FrontEnd

This section describes the tools that will be used at the front-end development of the application (client-side).

#### 3.1.1 Material Design

Material Design (codenamed Quantum Paper) is a design language developed in 2014 by Google. According to the Google website for Material Design, it's goal is to blend many principles of classic design with the innovations we've come to expect from science and technology.

Conceptually, Material Design is based on just that—real world materials. Using shadows and lighting to create depth and edges, elements respond in a similar way to how we would expect them to in real life. Again, Google said that Material Design is based on paper and ink and print-based design.

#### 3.1.2 Javascript

JavaScript (JS) is a high-level, interpreted programming language. [1] It was initially part of the implementation of web browsers, so that client-side scripts can communicate with the user, exchange data asynchronously, and dynamically change the contents of the document being displayed.

## Typescript

TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript, and adds optional static typing to the language.

## Angular

Angular is an open source, TypeScript-based front-end web application framework led by Google's angular team and community. Angular is a complete rewrite from the same team that created AngularJS (commonly referred to as "Angular.js" or "AngularJS 1.X").

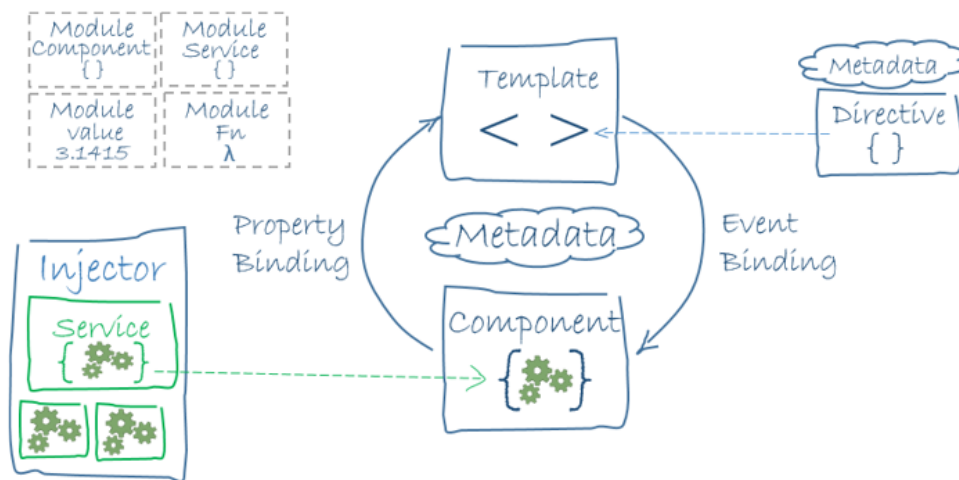


Figure 3.1: Angular Architecture overview

## DashJs

dash.js is an open-source client - side Javascript library for building video and audio players that playback MPEG-DASH content leveraging the Media Source Extensions API set as defined by the W3C.

### 3.1.3 Service Workers

Service workers are acting as proxy servers that are located between Web applications, the browser and the network (if available). They are designed in order to provide offline experience on the web, take appropriate actions based network availability or not and intercept network requests. Service workers also allow access to push notifications, background sync APIs and will manage some application caching capabilities.

## 3.2 Backend

This section describes the tools that will be used at the back-end development of the application (server-side).

### 3.2.1 Apache

Apache is a widely used open source HTTP server. A wide variety of functionality is supported and many of them are implemented as compiled modules that extend the core features of Apache. These can vary from server-side programming language to authentication schemes.

### 3.2.2 PHP

PHP is an open-source scripting language that is mostly suitable for web development. The original goal of this language was to implement a flexible templating system for HTML documents but nowadays it has been evolved into a general-purpose language.

### Laravel

Laravel is an open-source, PHP MVC web framework which provides the basic tools for building this thesis application.

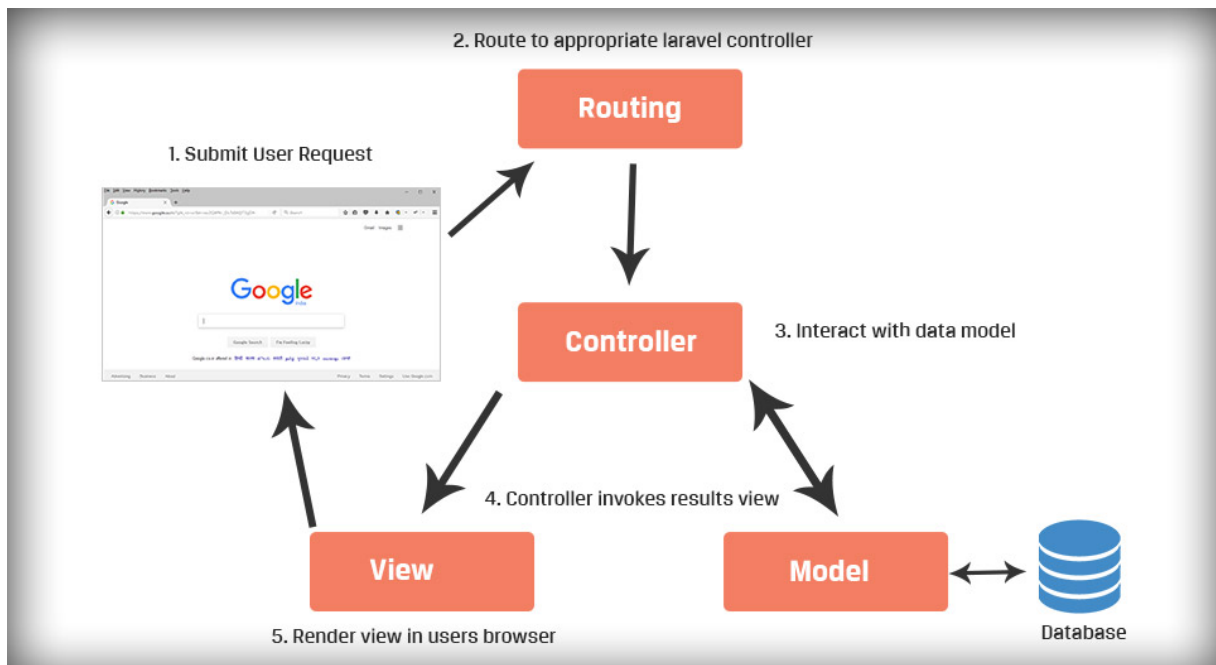


Figure 3.2: Laravel Architecture overview

### **3.2.3 Storage**

Amazon's S3 is a PAAS service for storing files in the cloud it provides high availability by replicating data across multiple servers within Amazon's data centers. S3 will be used for storing the assets.

### **3.2.4 Database**

#### **SQL**

Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation, initially based on relational algebra. The language includes recovery capabilities and data update, create and edit schemas and relational tables and access control over data.

#### **Mariadb**

MariaDB is a free open-source fork of the MySQL relational database management system. MariaDB offers master-master, master-slave replication and the Galera Cluster for multi-master deployments. MariaDB will store the relational data of the application

#### **K-V Store**

A database of key-values, or key-value store, is a data storage paradigm designed for the storage, retrieval and management of associative array, a data structure more commonly known today as a dictionary or hash.

#### **Redis**

Redis is an open source memory database project that implements an archive of distributed and in-memory key values with an optional duration. Redis supports different types of abstract data structures, such as strings, lists, maps, ordered sets, hyperloglogs, bitmaps and spatial indexes. This thesis is using Redis in order to manage user's session and managing experience points.

### **3.2.5 Video Streaming**

#### **H.264/MPEG-4 AVC**

H.264 or MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) is a video compression standard which can achieve high quality video in relatively low bitrates.

## Ffmpeg

FFmpeg is a set of free open source libraries, which allows the user decoding, encoding, transcoding, demuxing, muxing, filtering, streaming and playing digital audio and video in various formats. The name comes from the name of the MPEG and FF expert group, which means fast forwarding. This thesis is using Ffmpeg in order to encode video into multiple resolutions and bitrates.

## MP4Box

MP4Box is a multimedia packager available in GPAC suite. MP4Box can be used to prepare files mainly for HTTP downloading or RTP streaming. Mp4Box is used in this thesis for DASH preparation of the encoded video.

### 3.2.6 Virtualization

#### Docker

Docker is an open source software platform that implements virtualization at the operating system level. Basically, Docker offers automated procedures for deploying applications in isolated user spaces called Software Containers. The software uses core Linux technologies such as cgroups and kernel namespaces to allow independent execution of the container software on the same operating system. This avoids the use of additional computing resources that a virtual machine would require. (client-side) Docker is also providing the

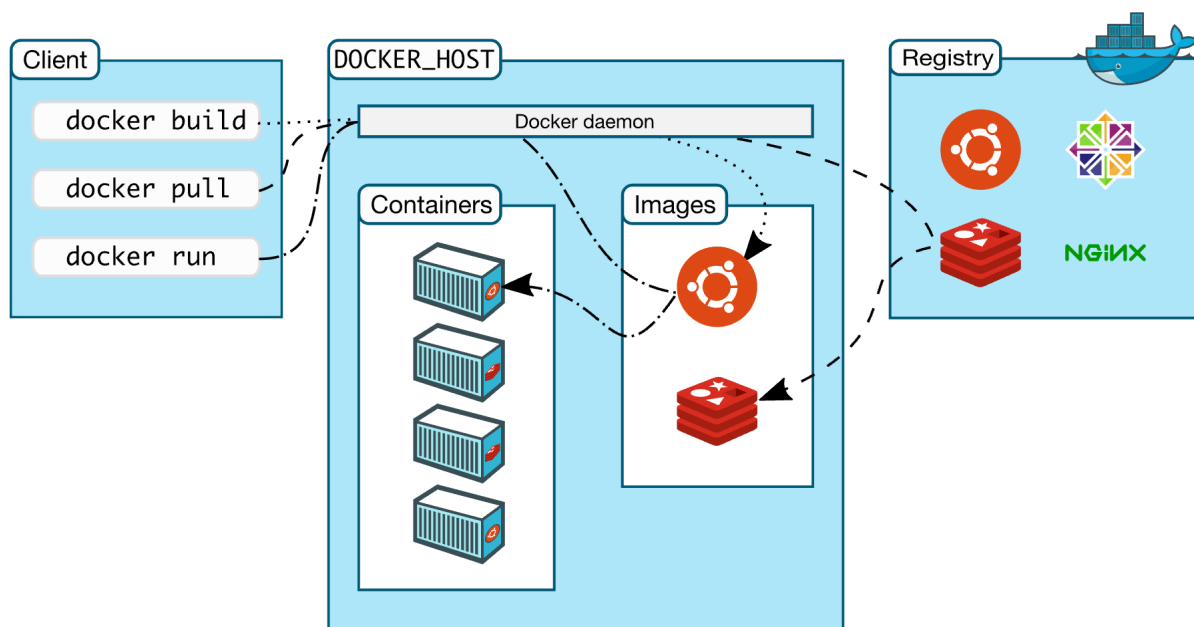


Figure 3.3: Docker Architecture Overview

following tools that are making software deployment easier and more flexible:

## Docker Compose

Docker Compose is a tool for defining and executing multi-container Docker applications

## Docker Swarm

Docker Swarm provides native clustering capabilities for Docker containers

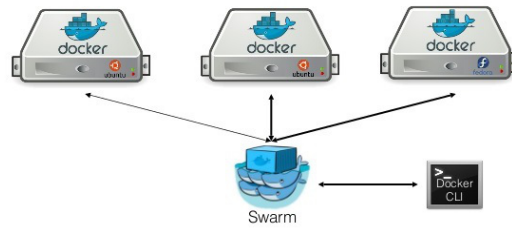


Figure 3.4: Docker Swarm overview



# Chapter 4

## Results

### 4.1 Authentication

#### 4.1.1 Auth Implementation

This thesis uses JWT for user authentication. After authentication, user's session data are stored in an in-memory k/v store (Redis).

#### 4.1.2 Login

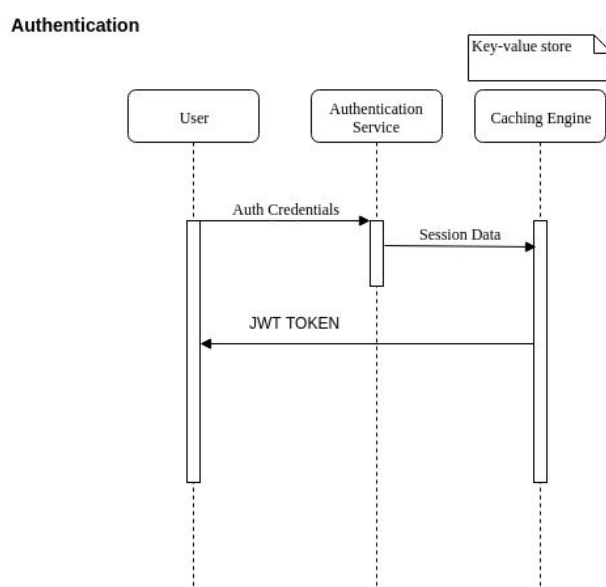


Figure 4.1: Authentication Architecture

#### 4.1.3 Register



Email  
**giannis@halvatzakis.me**

Password  
.....

**Login**

**Register**

Figure 4.2: Login Implementation

Full Name

---

Organization Name

---

Username

---

Email

---

Password

---

Password Confirmation

---

[Login](#)

[Register](#)

Figure 4.3: Register Implementation

## 4.2 Course Creation

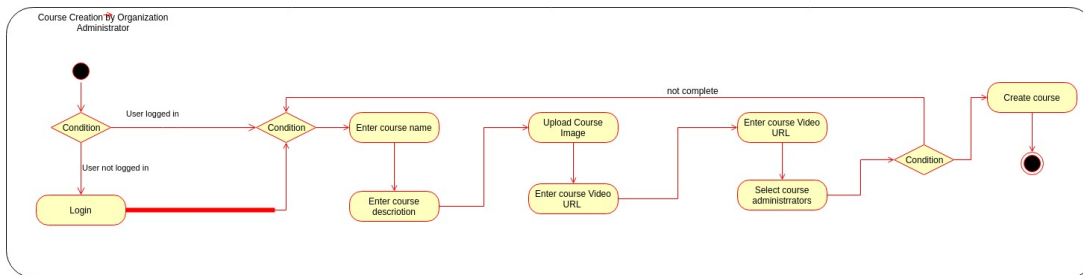


Figure 4.4: Course Creation

### 4.2.1 Course Details

The Administrator can add some course details that will mostly appear on the "Course Enrollment Section (Figure 4.15)"

The screenshot shows the "eLearning" interface with a user profile "xp:281". The "New Course" section features a progress bar with four steps: 1. General Course Details (active), 2. Lesson Designer, 3. Quiz Designer, and 4. Confirm. The form includes input fields for "Course Title", "Short Description", "Description" (with a rich text editor icon), "What Will You Learn" (with a rich text editor icon), "Year Created", and a "Categories" dropdown menu. A "Next" button is located at the bottom.

Figure 4.5: Course Details Implementation

### 4.2.2 Lesson Builder

In this section the Administrator can add lessons to the course, there are two types of lesson:

- HTML where the teacher can add plain html content.
- Video where the teacher can upload a video.

elearning

## New Course

1 General Course Details
2 Lesson Designer

---

<b>1). First Lesson</b>	First Lesson Description
<b>2). Second Lesson</b>	Second Lesson Description

---

<b>3). Third Lesson</b>	Third Lesson Description
-------------------------	--------------------------

Title  
Third Lesson

---

Description  
Third Lesson Description

---

Lesson Type

---

remove lesson

---

**New Lesson**

Back
Next

Figure 4.6: Lesson Creation Implementation

### 4.2.3 Quiz Builder

Using the Quiz Builder the teacher can create various adaptive quizzes. The teacher is able to select the questions, the difficulty of each question and the question answer types such as

- 2 choices, True or False selection
- Radio button group type
- Multiple choice with checkboxes
- Number selection between 2 numbers

### New Course

- 1 General Course Details
- 2 Lesson Designer
- 3 Quiz Designer
- 4 Confirm

1) First Quiz First Quiz Description

Title  
First Quiz

Description  
First Quiz Description

Attempt Time (in seconds)  
120

1) First Question

Question  
First Question

Difficulty Level  
0

Question Type  
multiple choice

1) Answer 1

2) Answer 2

Answer  
Answer 2  Right Answer

Remove Answer

Add Answer

Remove Question

2) Second Question

Add Question

Remove Quiz

2) Second Quiz Second Quiz Description

Add Quiz

Back Next

Figure 4.7: Quiz Creation Overview

**2).**

title

---

Description

---

Attempt Time (in seconds)

---

**Remove Quiz**

---

**Add Quiz**

Figure 4.8: Quiz Creation Fields

**1).**

Question

---

Difficulty Level

---

Question Type

---

**Remove Question**

**Add Question**  
**Remove Quiz**

Figure 4.9: Quiz Question Fields



## 4.3 Course Browser

The user can browse the available courses by pressing the browse courses button on the left sidebar. When navigates in the browse courses section, the courses appear in cards with basic information and the user can apply some category filters or searching on the top side of the screen.

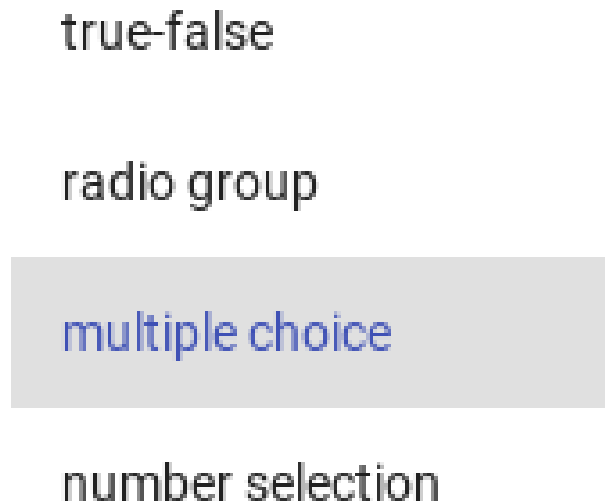


Figure 4.10: Quiz Question Types Selection

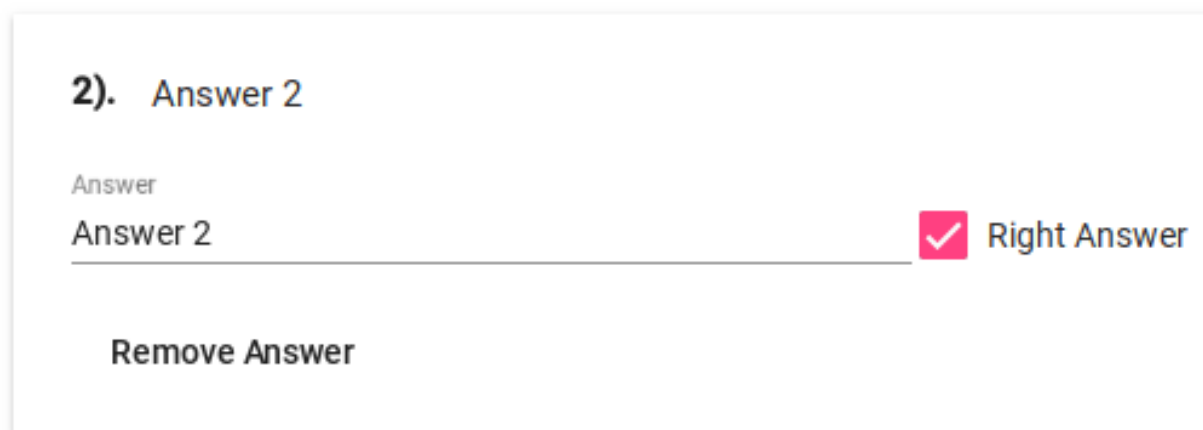


Figure 4.11: Quiz Question Answer

## 4.4 Course Enrollment

In the Course Enrollment section the user can see information regarding the course such as a basic background image with title and subtitle, long course description, an introduction

video, and a "what will you learn" section in order to inform the user about the course expectations. At the bottom side of the page there is a enroll button that registers the user into the selected course.

User Enrolls to Course

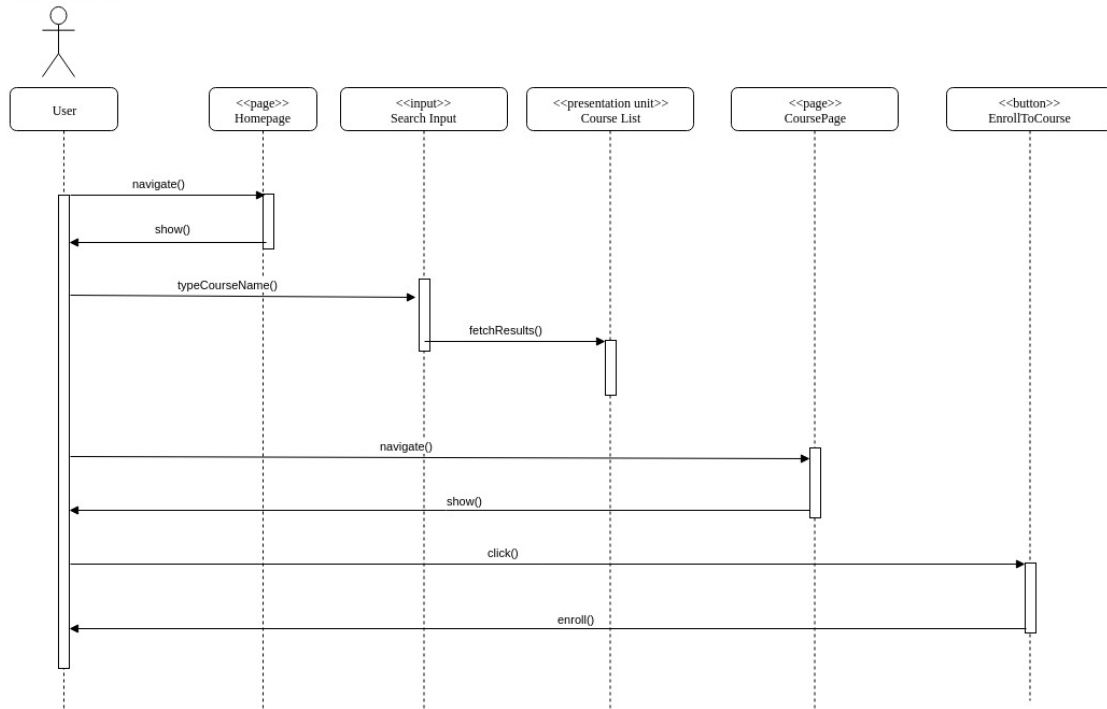


Figure 4.12: Course Enrollment Sequence Diagram

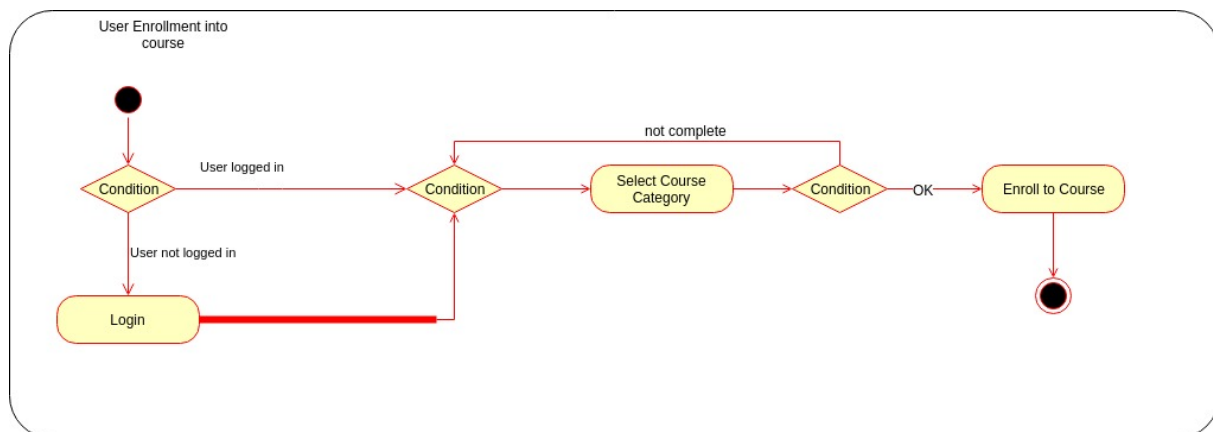




Figure 4.13: Course Enrollment Activity Diagram

### 4.4.1 Course Dashboard


In the Course Dashboard the user can see his progress, browse the past lessons and take adaptive quizzes.

elearning xp:471  

- cumque velit
- distinctio perferendis
- sint repudiandae
- eos praesentium
- cum odio
- vero in
- enim qui

CLEAR FILTERS


Vitae aspernatur iste.



Voluptatem voluptas natus et harum nobis ducimus ducimus voluptates rem.

ENROLL


Facilis ducimus suscipit numquam harum repellat.



Quisquam aut perspiciatis excepturi iste voluptate sed dolores ut alias soluta est sequi.

ENROLL

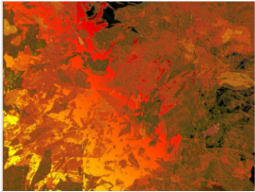
Voluptas officiis laboriosam dolor modi.



Aut ea ut facere vitae quam sapiente ab incidunt voluptas rerum.

ENROLL


Assumenda quo dolores et laborum dolorem.



Vero minima quo aut sunt aut consequatur consequatur nostrum doloribus consequatur voluptas.

ENROLL


Et assumenda a nostrum.



Dolorem magni est minima vitae perferendis dignissimos sint qui.

ENROLL

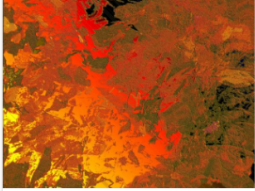
Perspiciatis voluptas autem ipsum.



Sint nesciunt vel soluta nobis voluptas hic necessitatibus quaerat sequi adipisci totam qui qui.

ENROLL


Esse perspiciatis architecto et itaque et.



Qui nihil quae quia nam amet voluptas et hic tenetur delentit.

ENROLL


Perferendis consequuntur neque.



Id non quisquam sit suscipit quod laboriosam quia esse numquam ducimus adipisci.

ENROLL


Consequuntur magnam quidem.



Voluptas quis ut quasi architecto consequatur molestiae voluptas.

ENROLL

Dolorem maiores perspiciatis alias ex.

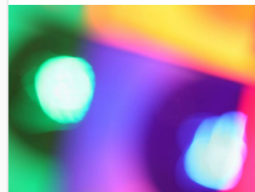


Laborum ut ut saepe quae quo laudantium inventore.

ENROLL

Giannis Halvatzakis

test



test

ENROLL

43

Figure 4.14: Course Browser

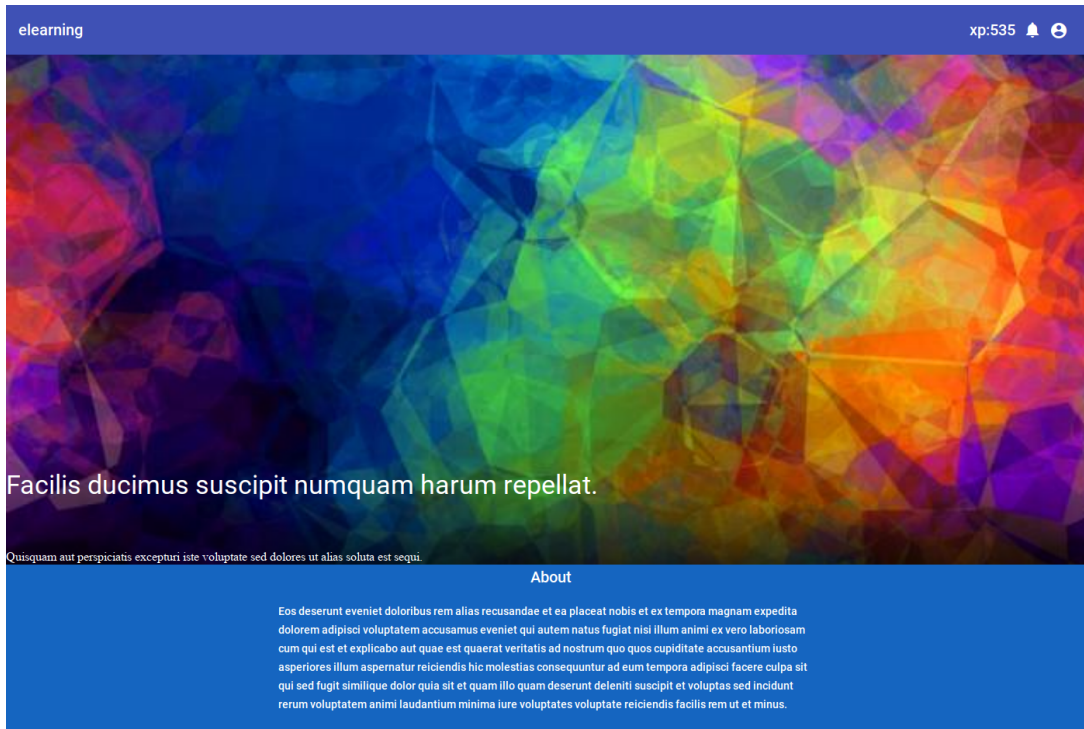


Figure 4.15: Course Enrollment Info

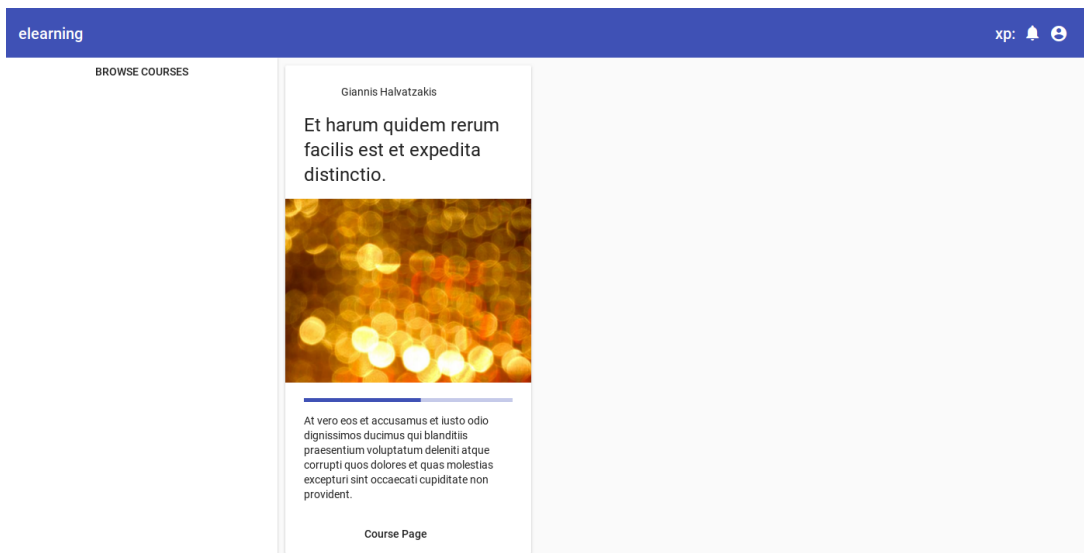


Figure 4.16: Enrolled Courses List

elearning xp: [notification icon] [refresh icon]

Facilis ducimus suscipit numquam harum repellat.

Lessons Watched: 9 of 43

Start

OVERVIEW LESSONS QUIZES OTHER

- Illo et illo.
- Sapiente quia est ipsum.
- Dolores quo laudantium earum.
- Fugiat facere ratione.
- Harum dignissimos culpa ad.
- Dolores ducimus voluptates nostrum.
- Cumque dolores id dignissimos qui expedita.
- Ipsa qui delectus consequatur qui.
- Non distinctio quo.
- Sequi numquam deleniti omnis tenetur amet.

Figure 4.17: Course Dashboard

## 4.4.2 Course Quizzes

In order to be evaluated the user has to take some adaptive quizzes. In an adaptive quiz the questions are randomly selected and the difficulty changes depending on the user's previous answers. If the previous answer was right, then the difficulty level increases and a most difficult random question is taken out of the question bank, if the user's answer was wrong then a more easy random question is obtained. When the questions in a difficulty level run out then the quiz ends and the user is earning experience points depending on the difficulty level reached.



Figure 4.18: Quiz Initial Screen

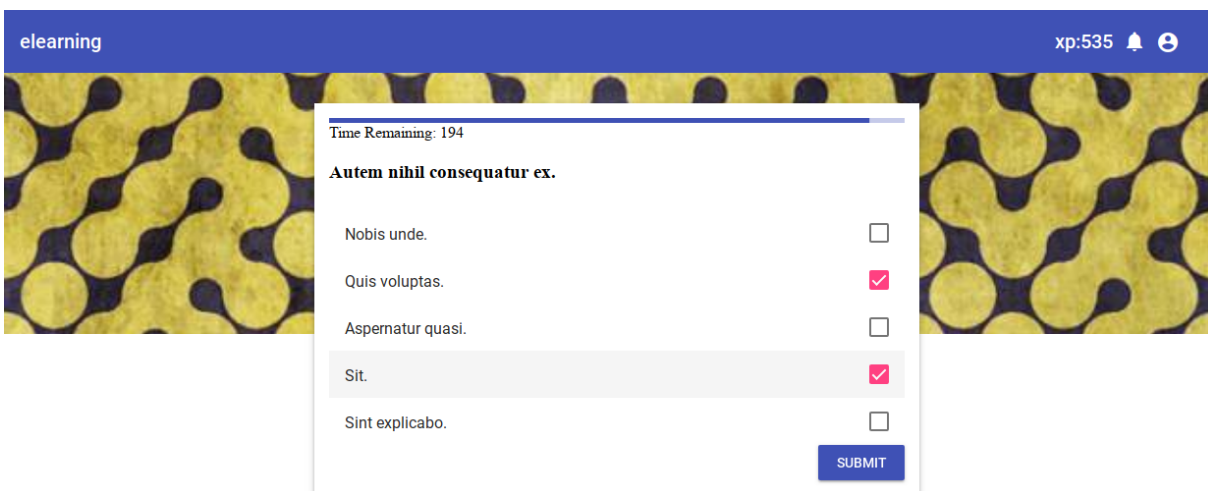


Figure 4.19: Multiple Choice Answers



Figure 4.20: True - False Answer



Figure 4.21: Quiz Results

## 4.5 Experience System

The experience system is a unit of measurement to quantify the user's progression through the e-learning platform. The user can gain experience points by taking some actions such as watching lessons or taking quizzes.



# Chapter 5

## Conclusions and Future work

This chapter contains the conclusion of this thesis and also future work and expansions for the E-Learning platform.

### 5.1 Conclusions

In this thesis we implemented a e-learning platform with various elements collected from the new learning system[8]. More specifically we focused on improving the assessment part by building a adaptive testing system where difficulty selection of the questions is based on the previous user's answers. By taking quizzes, user is rewarded with experience points which is part of a general reward system.

As regards the front-end part of the application, Javascript (Angular) was used along with angular material for building the ui part.

For the back-end part of the application, PHP (Laravel Framework) was used for building the RESTful api, Authorization and data management.

Several services such as Ffmpeg, Redis, Mariadb, and the multimedia packager are running through containers (docker) and for the storage part Amazon's S3 was used.

### 5.2 Future work

The E-Learning platform has a lot of room for optimizations and improvements. First of all a global content delivery network (CDN) can be used for data delivery that provides low latency and high transfer speeds globally. Load balancing can be easily achieved for the Docker services by implementing a HAProxy solution. Furthermore Google's Kubernetes can be used for automatic cluster management.



# Bibliography

- [1] J.T. Rayfield A. Leff. Web-application development using the model/view/controller design pattern, August 2002. Cited on page 24.
- [2] Gerard Briscoe and Alexandros Marinos. Digital ecosystems in the clouds: towards community cloud computing, 2009. Cited on page 21.
- [3] Cedefop. Terminology of european education and training policy: a selection of 130 terms, November 2014. Cited on page 16.
- [4] Stefan Hrastinski. Asynchronous and synchronous e-learning, November 2008. Cited on page 16.
- [5] Michael Moore & Greg Kearsley. Distance education: a systems view, November 2005. Cited on page 16.
- [6] Ramesh K. Sitaraman Kevin Spiteri, Rahul Urgaonkar. Bola: Near-optimal bitrate adaptation for online videos, January 2016. Cited on page 25.
- [7] Niels Laukens. Adaptive streaming: a brief tutorial, 2011. Cited on page 25.
- [8] Bill Cope Mary Kalantzis. New learning: Elements of a science of education 2nd edition, June 2012. Cited on pages 9, 13, 17, and 49.
- [9] Peter Mell and Tim Grance. The nist definition of cloud computing, July 2009. Cited on page 20.
- [10] G. G. Weiss, D. J.; Kingsbury. Application of computerized adaptive testing to educational problems, December 1984. Cited on page 19.



# Appendix A

## Component Diagram of the E-Learning platform

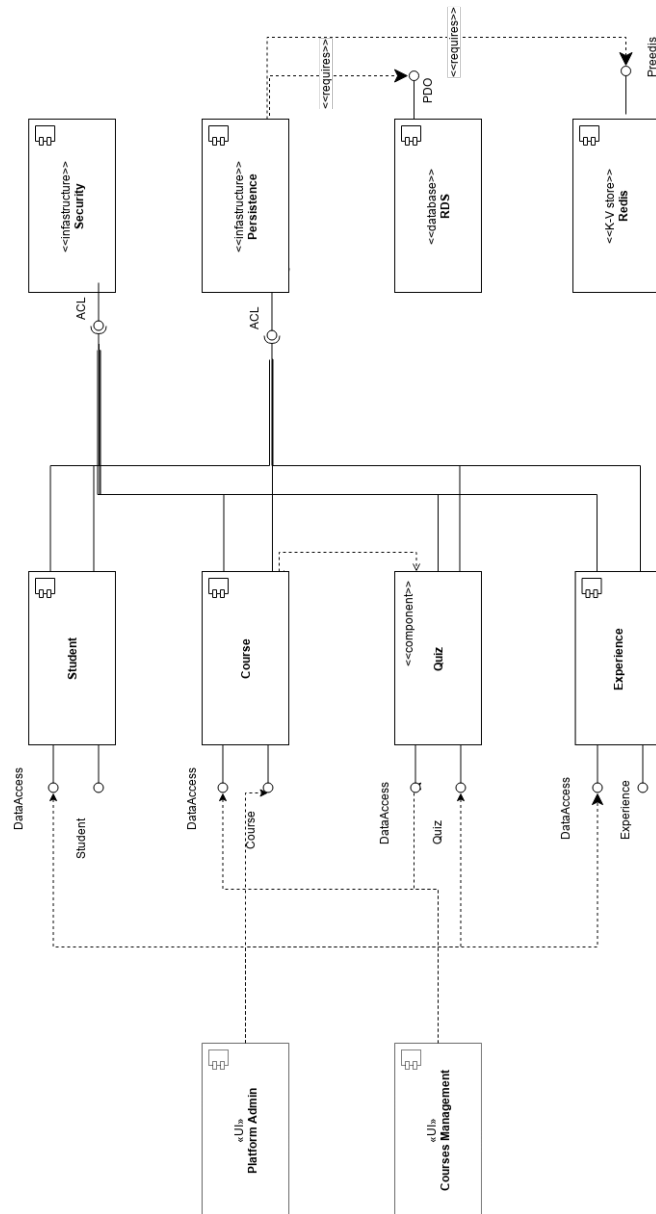


Figure 5.1: \*

# Database Schema of the E-Learning platform

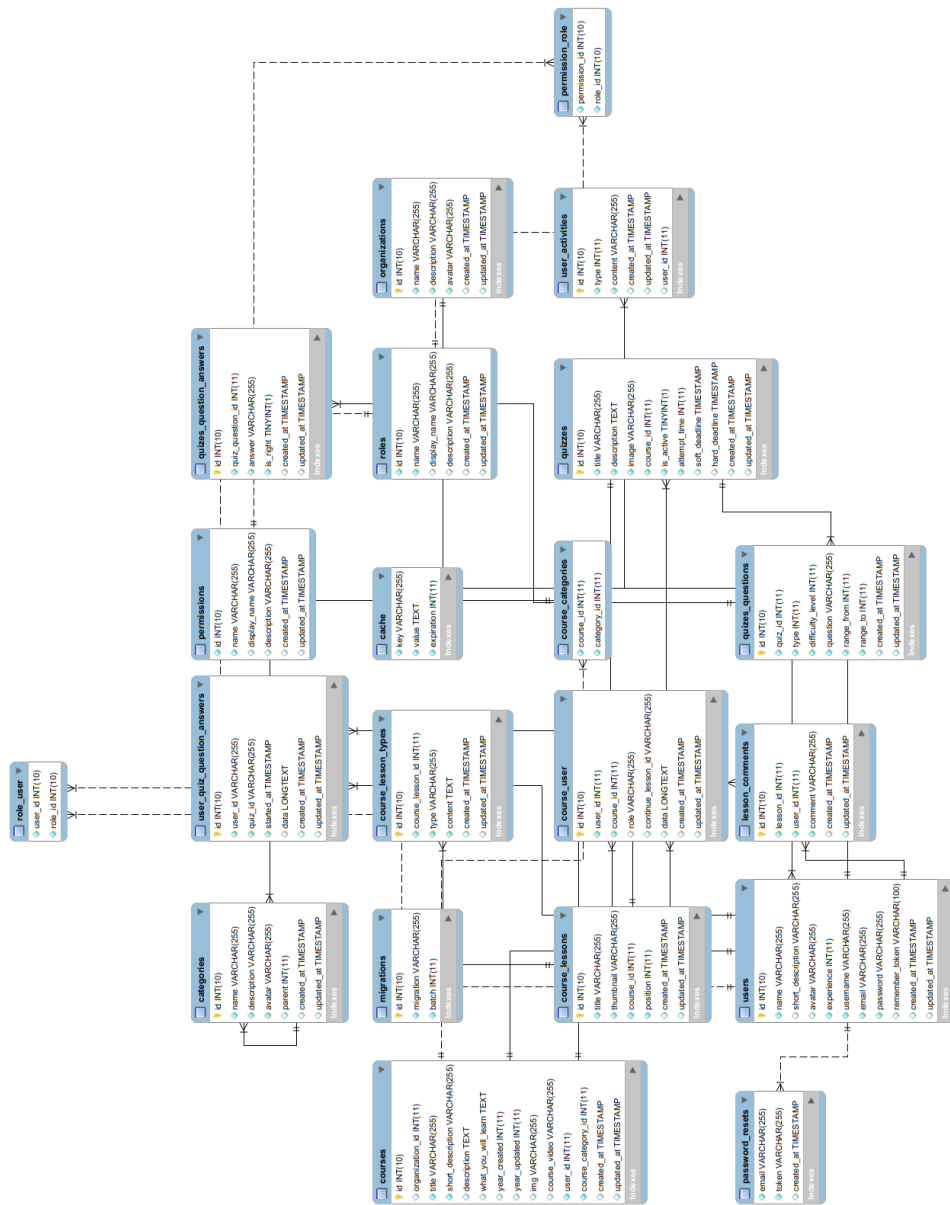


Figure 5.2: \*

# Class Diagram of the E-Learning platform

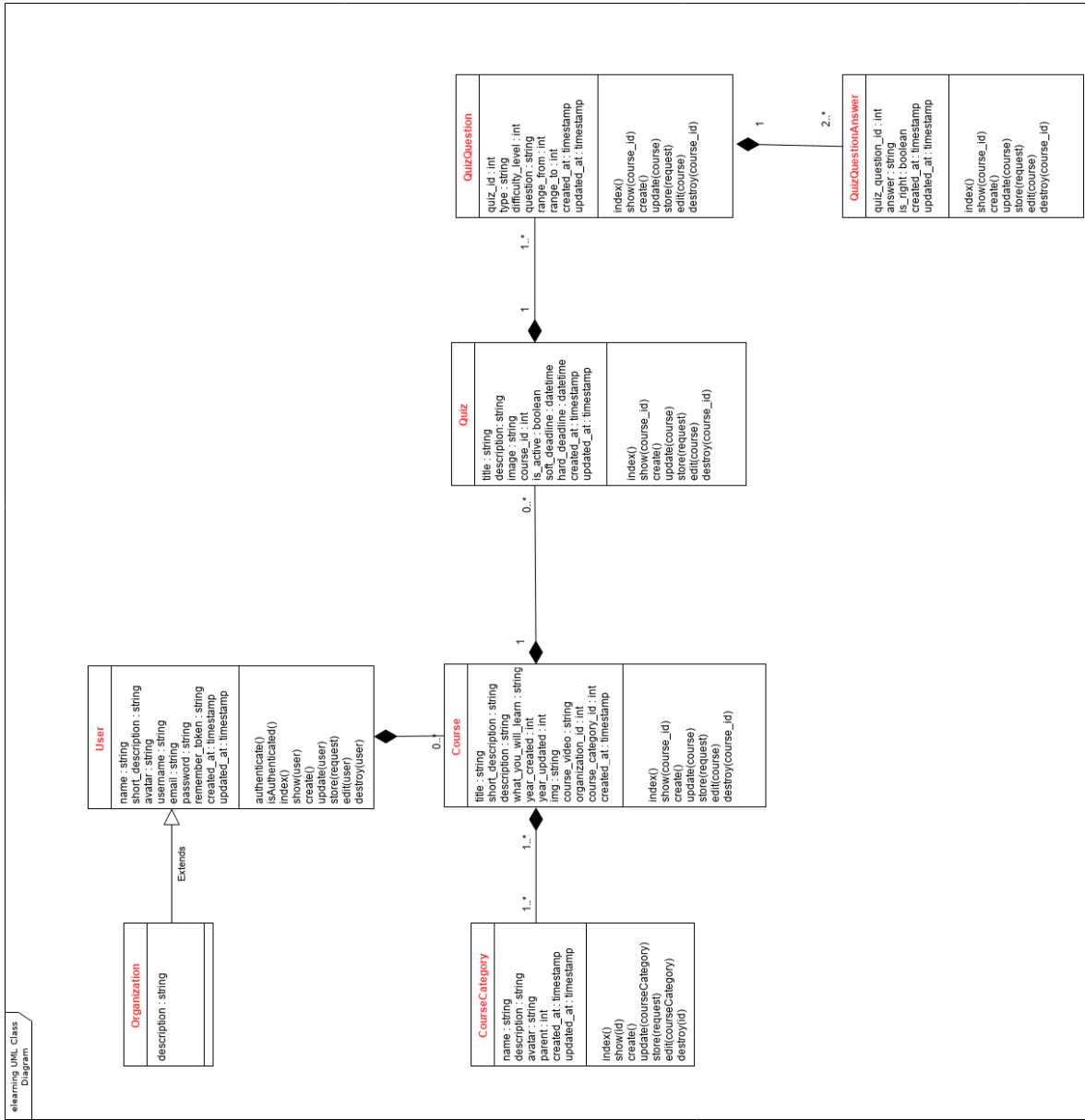


Figure 5.3: \*

