LAB University of
Applied Sciences

DEVELOPING A LEARNING MANAGEMENT SYSTEM USING SINGLE PAGE APPLICATION ARCHITECTURE

LAB UNIVERSITY OF APPLIED
SCIENCES LTD
Bachelor of Business Administration
Degree programme in Business
Information Technology
Autumn 2020
Ivan Leontev

Abstract

Author(s) Leontev, Ivan	Type of publication Bachelor’s thesis	Published Autumn 2020
	Number of pages 43	

Title of publication

Developing a learning management system using single page application architecture

Name of Degree

Bachelor of Business Administration

Abstract

Development of IT industry is one of the factors that have a very strong impact on education, especially through E-learning systems. Such systems became even more important in 2020 for the education sector, due to the COVID-19 pandemic.

The aim of this thesis was to describe the development process of a learning management system platform and in order to achieve this goal, the process of single-page application development was shown. This paper was divided into three main stages: planning, risk analysis and requirements, and design and prototyping. In addition, the thesis aimed to explore the MVC pattern and provide a list of suggestions and recommendations for further system improvement.

The practical result of the thesis was a functional single-page application prototype that met all the requirements formulated during the design stage of the system.

Keywords

learning management system, single page application, development process

CONTENTS

LIST OF ABBREVIATIONS

API	Application Programming Interface
CSS	Cascading Style Sheets
DBMS	Database Management System
DOM	Document Object Model
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
JS	JavaScript
LDAP	Lightweight Directory Access Protocol
MVC	Model-View-Controller
OOP	Object-Oriented Programming
ORM	Object-Relational Mapping
PDO	PHP Data Objects
PHP	Hypertext Preprocessor
POP3	Post Office Protocol Version 3
SDLC	Software Development Life Cycle
SPA	Single-Page Application
SQL	Structured Query Language
WYSIWYG	What You See Is What You Get

1 INTRODUCTION

1.1 Research Background

Education plays a crucial role among social institutions of modern society. Long-term rates of economic, political, and social development of the state depend on the quality of education. For this reason, the education sector should always be at the forefront of innovative development. (Ponchon 2015)

In this regard, educational institutions always strive to apply new approaches and technologies in teaching. One of the tasks of this development is the creation of online learning environments that contain training materials and manuals for additional study purposes. The relevance of this activity lies in the fact that students receive an additional incentive to self-education and creative development. (ED 2011)

It is also important to mention that in early 2020, due to the COVID-19 outbreak, educational institutions around the world have temporarily switched to distance learning. This experience has shown that the existing distance learning tools cannot always cope with the challenges they face. On the one hand, there may be difficulties with mastering the learning management systems (Plitnichenko 2020). On the other hand, as the recent case of Zoom has shown, cybersecurity and data privacy issues (Harwell 2020).

The development of a single-page application for an online learning environment that solves this range of problems requires a number of modern technological and architectural solutions. It is enough to use a combination of HTML and CSS to create a template and design of the application, but to write the functional part it is necessary to use fully-fledged programming languages such as PHP and JavaScript. It is also important to use frameworks such as AngularJS and PHPixe to ensure that the application under development meets modern standards and the MVC pattern.

1.2 Research Objectives, Questions and Methods

The main objective of this research is to develop a single-page application that will be a platform for learning management system.

In addition, the thesis aims to:

- Explore the MVC pattern
- Explain the application development process
- Provide a list of suggestions and recommendations for further system improvement

Research question and sub-questions help to bring focus to this thesis and provide a clear structure.

The main research question for this thesis is:

- What is the process of developing a single-page application for an online learning environment?

In order to provide a structured answer to the research question, seven sub-questions were specified:

- What is the architecture of the application under development?
- Which tools and technologies to use to build the system?
- What are the existing solutions on the market?
- What is the system's life cycle?
- What are the system requirements?
- What are the data structure requirements for the system?
- Which sections are available in the system?

The practical result of the thesis is a functional single-page application that meets all the requirements formulated during the design stage of the system.

The following research methods are applied for writing this thesis:

- Special literature review
- Competitor analysis
- Prototype development

1.3 Thesis Structure

The thesis consists of five main sections. Figure 1 below illustrates the overall structure of the thesis.

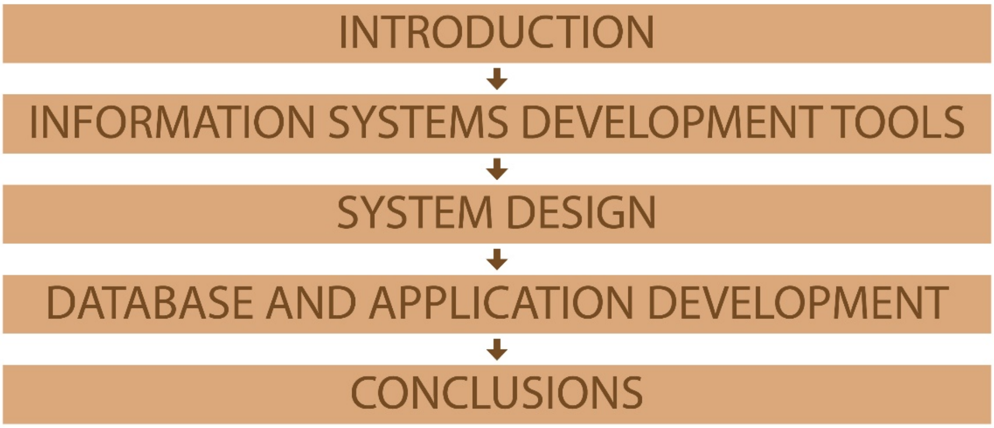

INTRODUCTION
INFORMATION SYSTEMS DEVELOPMENT TOOLS
SYSTEM DESIGN
DATABASE AND APPLICATION DEVELOPMENT
CONCLUSIONS

Figure 1. Thesis structure

Chapter 1 reveals the relevance, objectives, and research methods. It also specifies research questions as well as explains the theoretical and practical importance of the work.

Chapter 2 focuses on the theoretical aspects of single-page application development. It explains the architecture of the SPA and all the corresponding technologies used in the system.

In Chapter 3, there is an analysis of existing solutions in the selected area. It discusses two products and defines their main features and functionalities. After that, it explores several life cycle models and defines one that is the most suitable for the system. Also, the chapter reviews the system requirements and data structure requirements. In this chapter, the future system and its accompanying database are also designed.

Chapter 4 covers directly the single-page application development. It discusses the implementation of the required functionality as well as the appearance of the application and available sections.

Chapter 5 concludes and summarises this thesis. It presents research results and gives the answers to the research questions. Finally, it contains suggestions and recommendations for further system improvement.

2 INFORMATION SYSTEMS DEVELOPMENT TOOLS

Before actually writing the application logic and developing its interface, it is necessary to consider the system architecture in detail. And it is also extremely important that the system is not only able to perform all the required functions, but also has a certain potential for extending and modifying the functionality.

2.1 Architecture: Front-End and Back-End Division

The back-end is the code responsible for application business logic and data handling. The front-end, on the other hand, is responsible for the client part of the project, application interface, and ways of interaction with it. Separating the project internal structure into front-end and back-end is useful for several reasons:

- The project becomes more scalable.
- Front-end and back-end developers are able to work independently of each other.
- The project code becomes cleaner and more logical.
- The same front-end and back-end parts can be reused in other projects. (GeeksforGeeks 2020)

In the case of the system being developed, front-end and back-end will be two independent applications working in pairs.

2.2 Architecture: MVC Design Pattern

If it is initially assumed that the project will have extensive functionality, then it is worth thinking about using patterns and frameworks. The use of patterns and frameworks will make the application flexible and well-structured, as well as accelerate the development. (Gamma, Vlissides, Helm & Johnson 1994, 12)

Model-View-Controller (MVC) is a fundamental programming pattern that separates the business logic of an application from its graphical interface. Originally, the MVC pattern was proposed in 1979 by the Norwegian developer Trygve Reenskaug, who at that time was working in Xerox Parc on the programming language Smalltalk. (Gamma et al. 1994, 14-15) Over time, the pattern has been finalized, but its concept remains unchanged. In web programming, the MVC pattern has become widely used only in recent years with the spread of such popular development frameworks as Ruby on Rails, Zend Framework, and Struts.

The concept of the pattern is quite simple: it is necessary to clearly separate the responsibilities of the components. In general, it means that the business logic should not be described in the view (GUI of the application) and the interface cannot be described in the model. As a result of applying the MVC pattern, the application is divided into three main components, each of which is responsible for different tasks. (Freeman, Robson, Bates & Sierra 2004, 530-531) Figure 2 below provides a general scheme of interaction between the pattern components.

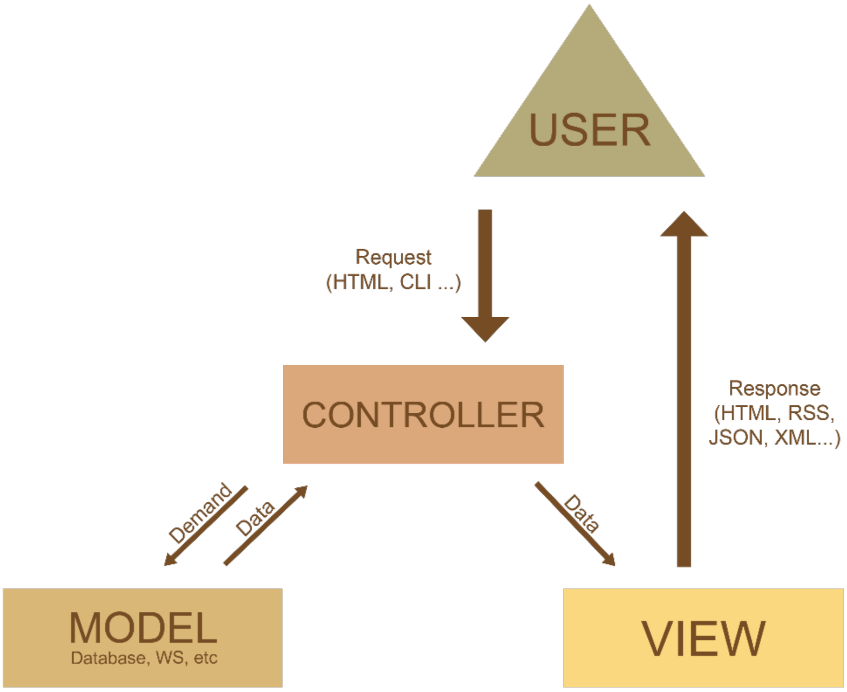


Figure 2. Conceptual diagram of the MVC pattern (adapted from Freeman et al. 2004)

As the name implies, the MVC pattern is based on three interrelated components: model, view, and controller. What tasks do they perform?

2.2.1 Model

The model contains business logic and processes data. It processes, validates, and changes data. Only the model can directly access the database or any other data source out of all application components. The model never interacts with the user and cannot visualize data. The main purpose of the model is to process the request and provide the data that will be displayed in the view.

2.2.2 View

The view is used to create the application interface. In the case of a web application, the view will contain HTML and CSS markup. The view cannot directly access the database or the model. It can use methods and properties from the controller for data output.

2.2.3 Controller

The controller is a linking element of the MVC pattern. The controller manages the user requests received by the GET and POST methods. It directs them to the models and invokes the necessary views. There are no HTML markup and data requests inside the controllers. (Freeman et al. 2004, 532)

As a result of using the MVC pattern, the presentation logic and application logic are separated from each other, which simplifies testing, debugging, and maintenance of the application, as well as further modification and code expansion.

2.3 Front-End

Any website or web application starts with appearance. As in real life, there is a “judge a book by its cover” rule. No matter how powerful and complex algorithms are at the core of the system – if the end user does not understand the interface, he or she will not need these algorithms. Therefore, one of the main and at the same time the most difficult tasks is interface development.

The complexity of developing an interface for the Web lies in the fact that today the Internet is available everywhere from any device: from a smartphone to a computer. Meanwhile, the final system interface should be as convenient as possible for users from any device.

Over the past few years, multiple solutions to this problem have been presented. Some time ago, for example, the development of separate mobile versions of websites was a popular solution. This is a good option, but it essentially requires double development costs and further support, which means that it is not very suitable. (Codesido 2009)

In this work, there will be used another approach – responsive web design. Markup of web pages of the application under development will be written in HTML language, and the appearance will be described in CSS language. Besides that, in order to simplify the interface development, there will be used the Twitter Bootstrap CSS framework. The use of Bootstrap allows creating a consistent and responsive design as quickly as possible, thus providing more time for system logic development.

2.3.1 HTML 5

HTML is a hypertext markup language used to describe the structure of web documents. HTML was proposed by the British scientist Tim Berners-Lee at the dawn of the Internet in 1990. In HTML documents the markup is described using special descriptors (tags), which are then interpreted by browser programs. HTML is constantly evolving, the latest version of the HTML standard (HTML 5.2) is recommended for use by the W3C consortium from 14 December 2017 (W3C 2017).

HTML 5 gave the web developer more freedom, for example, there is support for geolocation, audio and video playback, new semantic tags that simplify the work with search engines, and much more (MacDonald 2013, 6). The so-called data attributes should be considered in more detail.

In HTML 5, with the help of the data-* prefix, it is possible to create custom attributes applicable to any tags, which will be valid in terms of HTML 5, as well as will be accepted by modern browsers. Data-* attributes allow to store any information, which can, for example, be transferred to JavaScript and modified. In fact, data attributes become sort of variables inside the DOM tree. The use of data attributes will be useful in the process of developing the system interface logic. (HTML Tutorial 2020)

2.3.2 CSS 3

CSS is a formal language for describing the appearance of a document. CSS always works in conjunction with the markup language. Initially, the language was developed to separate the template logic from its presentation and thus to unload HTML (or any other markup language). As with HTML, the CSS language is evolving. Its first version, CSS1, was released in 1996 and the latest version, CSS3, is still under development although CSS3 elements are already widely used by web developers. (Meyer 2000, 8)

CSS3 should bring the most extensive changes to the specification since the language was introduced. The third version of the specifications allows to add shadows, gradients to the elements, and add animations to them without using JS. Another major innovation is Media Queries.

Media Queries is the cornerstone of responsive web design. Today, the screen size of devices varies within a very wide range. It is almost impossible to create a web page that will look good in all this range using traditional web development tools. Until recently developers either created mobile applications or separate versions of websites for mobile devices. (Gilbertson 2012) The Media Queries module can slightly simplify the life of a web developer. For example, by using Media Queries the developer can specify that some rules should only be applied on devices with screen resolution less than 920px. Thus, the developer is able to create responsive web design only by means of CSS, hiding or moving blocks on the screen. (CSS Tutorial 2020)

2.3.3 Twitter Bootstrap

Twitter Bootstrap (or simply Bootstrap) is an open-source interface framework that contains HTML and CSS design templates for typography, web forms, buttons, labels, navigation blocks, and other design components.

The advantage of using CSS frameworks is that the layout designer does not need to think about the many nuances of the layout that the framework creators have already considered for them. These nuances include cross-browser compatibility, support for different screen resolutions, and much more. The layout designer only specifies what, how, and when it should be displayed, the rest the framework does itself. This approach can significantly accelerate the layout process of the website.

Bootstrap makes the development of responsive design even easier. In Bootstrap, there are rules for displaying content on regular and large monitors, smartphones, and tablets. Besides that, Bootstrap provides an opportunity to hide and show separate blocks on different devices. (Bootstrap 3 Tutorial 2020)

2.3.4 AngularJS Framework

AngularJS is an open-source front-end JavaScript framework. It was originally developed by Google and it is used to create single-page applications as well as to interact with DOM. AngularJS is based on the MVC pattern – it divides the application into three separate parts that can be changed independently of each other. This approach makes it possible to write and test code quickly and easily. (AngularJS Tutorial 2020) AngularJS is also a great combination of a low entry threshold and a rich feature set, which makes it an excellent framework for the system under development.

2.4 Back-End

The application business logic must be concealed from users. Therefore, the application logic has to be executed on the server-side and the PHP language is well suited for this purpose. Data are also needed in the model. Since the data in the project will be dynamic, it will be stored in a relational database.

2.4.1 PHP

PHP is a hypertext preprocessor, a scripting programming language, which is used for web application development. The PHP language is C-like, with some elements (like foreach loop) from Perl. PHP was developed by Danish programmer Rasmus Lerdorf in 1994. (PHP Manual 2020)

PHP is a server-side technology, which means that PHP scripts are executed exclusively on the server-side, as opposed to JavaScript, for example. This allows to effectively use PHP to write the back-end part of the application. (Beighley & Morrison 2009, 13)

PHP also supports a wide range of databases and can interact with external services (LDAP, IMAP, POP3, HTTP, etc.). (PHP Manual 2020)

PHP scripts can be located inside HTML code. When PHP starts processing the file it will execute all the PHP code found inside and then output the HTML. However, when PHP is only needed to implement business logic, it can act as the application's API. Thus, it implements the model in the MVC pattern. In this role PHP will not generate pages, it will only receive requests and send responses to the front-end. (Zandstra 2004, 317-318)

2.4.2 PHPixie

PHPixie is a small open-source PHP framework that uses the MVC pattern. The PHPixie caching module reduces the load on the database by saving the results of frequently executed queries on the server. There is also an ORM technology implemented in PHPixie. ORM (Object-relational mapping) is a programming technology that links OOP concepts with relational database management. ORM has several advantages compared to plain-SQL queries:

- Conciseness – in ORM, the “SELECT * FROM clients” query can be executed by writing “Clients::objects->all()”.
- Universality – different DBMS have their own nuances of SQL query processing. If at some point the developer decides to change the DBMS, it is likely that he or she will have to rewrite most of the queries to it. In ORM it is only necessary to change the database driver.
- Object generation – as a result of the query, the ORM returns not just an array, but an object with methods and properties of its own class. (PHPixie Documentation 2020)

2.4.3 MySQL

SQL is a structured query language. A formal programming language used to create, manage, and modify data in relational databases. ANSI and ISO organizations are responsible for SQL standardization, but most DBMS extends SQL by creating their own dialects. This practice leads to compatibility problems between different DBMS.

MySQL is most frequently used as a DBMS on websites and in web applications. The popularity of MySQL in web projects is caused by the fact that the DBMS was originally developed in order to provide high performance, MySQL support is included in the standard PHP package, and finally, MySQL is free, which makes it an excellent choice for small and medium projects. (MySQL 8.0 Reference Manual 2020)

PHP supports working with MySQL. In addition, the PHP+MySQL tandem is one of the high-performance ones. In PHP there are three APIs to access MySQL databases in parallel: ext/mysqli, PDO_MySQL, and ext/mysql. The latter API is considered obsolete and will be removed in future versions of PHP. (PHP Manual 2020) The Mysql_PDO extension will be used in this work.

PDO is a layer for working with several databases. It allows developers to change the database driver without rewriting queries to the database where the SQL syntax may differ. PDO also supports prepared statements, which strengthen protection against SQL injection attacks. (PHP Manual 2020)

3 SYSTEM DESIGN

The design of the information system is an important stage of its development. At this stage, it is necessary to select the systems development life cycle, develop the system concept, and terms of reference. In addition, at this point, it is also advisable to analyze existing products on the market that solve similar tasks.

3.1 Analysis of Existing Solutions

Nowadays there are many solutions for creating online teachers' rooms and course management systems. It is important to analyze them and identify their advantages and disadvantages, which will help in the design and development of the system.

3.1.1 Moodle

Moodle is a free and open-source course management system (under the GNU GPL). Thanks to its distribution model, Moodle has become widespread in the world. Among the analyzed systems, Moodle is the only system distributed free of charge, which is its absolute advantage. (About Moodle FAQ 2019)

Moodle contains powerful tools for creating and managing courses. There are dozens of tools available for teachers to create an interactive and interesting environment for students. However, this is also one of the main disadvantages of Moodle – the difficulty of mastering. All new features that appear in new versions of the project, over time, have overloaded the interface to the extent that today in order to fully master all the features of the system teachers have to either spend many hours of their personal time or read specially published books or take expensive courses.

The openness of the system provides space for modification (under the terms of the GNU license). There are many additional modules on the Internet that extend the capabilities of the system or modify it to meet the needs of specific educational institutions.

Moodle supports any platform that supports PHP (Linux, Windows, macOS) and works with all popular databases including MySQL, MS SQL, PostgreSQL, Interbase, FoxPro, Access, ADO, and some others. (Moodle architecture 2018)

Furthermore, the system provides ample opportunities for communication. The exchange of files of any type is supported, and there are several ways to organize feedback: through forums, chats, and personal messages.

In general, Moodle is a very powerful system that retains its flexibility through its modularity expressed in a title. Open source provides a high level of security, and a large community can help with difficulties. The only major disadvantage is the entry threshold. In order for Moodle to reach its full potential, the organization will have to conduct certain activities with both teachers and students.

3.1.2 WebCT

WebCT is a proprietary course management system of Canadian origin. In contrast to other systems of its class, WebCT is implemented using Java Servlet technology. This implementation is both an advantage and a disadvantage of the system under consideration. Java is a cross-platform technology, which means that the system will work on any device with a browser and Java installed. At the same time, non-rational use of Java can have a very strong impact on system performance. (WebCT Java Problems 2008)

Another major drawback of WebCT is the pricing policy. WebCT will cost the customer $15,000 in the cheapest package (Angelo 2002). In addition, the system has a number of limitations on the number of users working simultaneously in the system. There will be an extra charge for exceeding this limit.

Functionally WebCT provides a similar set of features to Moodle, which makes this system quite a powerful alternative to Moodle. Besides, it also has official technical support.

3.1.3 Analysis Results

Based on the results of the analysis of the two e-learning systems, it can be said that, in general, all of them provide approximately the same functionality, significantly different in technical aspects of implementation and ways of providing access to the product. WebCT has a lot of possibilities, but the pricing policy may deter potential customers. Moodle is a low-cost solution to the problem of distance learning program implementation. Although Moodle will not require a financial contribution from the institution, instead it will require a certain amount of manpower to deploy a complete system. The Moodle deployment may also incur indirect costs for staff training and system deployment itself.

The analysis showed that these two products are powerful representatives of their class, but often the speed of deployment, ease of use, and usability are more important than the countless features. So far there is no such system on the market at the moment, which means it can be developed. The new solution must be modular, adaptive, user-friendly, and not resource demanding.

3.2 Life Cycle

Systems development life cycle (SDLC) is a period of time from the moment when the decision on system development was made until its complete decommissioning. The life cycle model determines which stages the system should go through and which processes are performed at each stage. The SDLC model depends on the specifics of the information system and the conditions under which the latter is created and operates. Over time, several general life cycle models have been proposed, which will be discussed below in this paper. (System Analysis and Design Tutorial 2020)

3.2.1 Waterfall Model

The waterfall model assumes that the whole development process looks like a flow that goes through several successive phases: requirements analysis, design, implementation, verification, and maintenance (Figure 3).

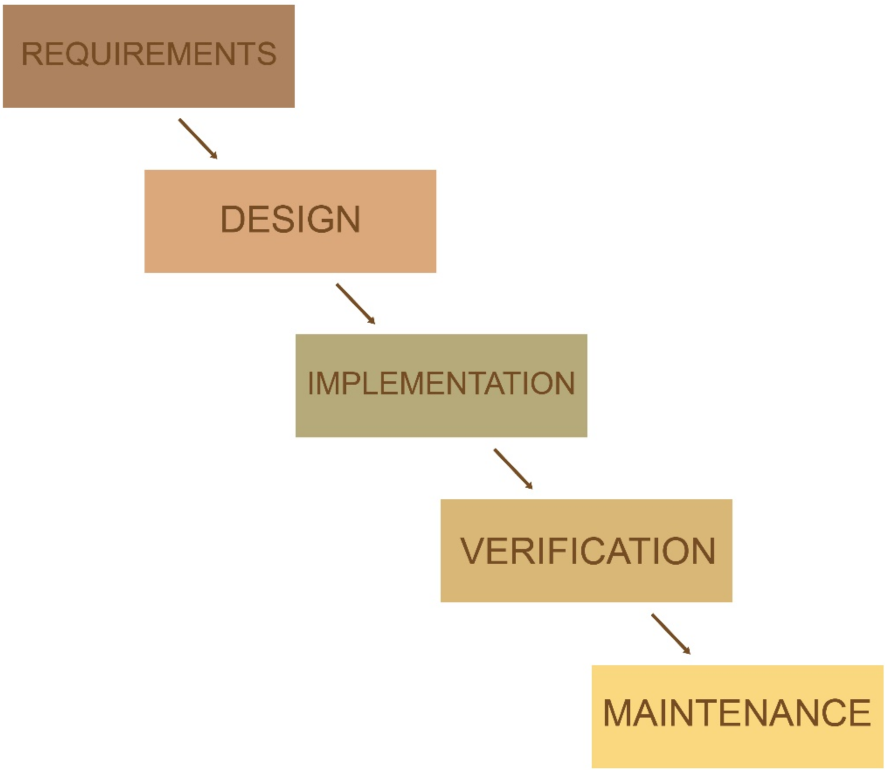


Figure 3. Waterfall model (adapted from Royce 1970)

This model implies that the developer has to complete the tasks of one phase to move on to the next. In the original version, the transition to the previous stages is not possible.

The obvious disadvantage of this model is the impossibility to return to the previous stage. In the real development environment, there is almost a constant need to change or refine the decisions taken earlier. (Oxagile 2014)

3.2.2 V-Model

V-model is a later variant of the waterfall model, it was named for its specific graphic representation (Figure 4). In V-model, development tasks are on the left side and testing tasks are on the right side. Iteration and return to the previous stages in this model are achieved by linking the tasks of the left part of the model to the right part.

The V-model has adopted some disadvantages of the waterfall model. Both models are poorly adapted to changes in customer requirements. In the case of long development times, the final result can differ significantly from the customer's needs. (Firesmith 2013)

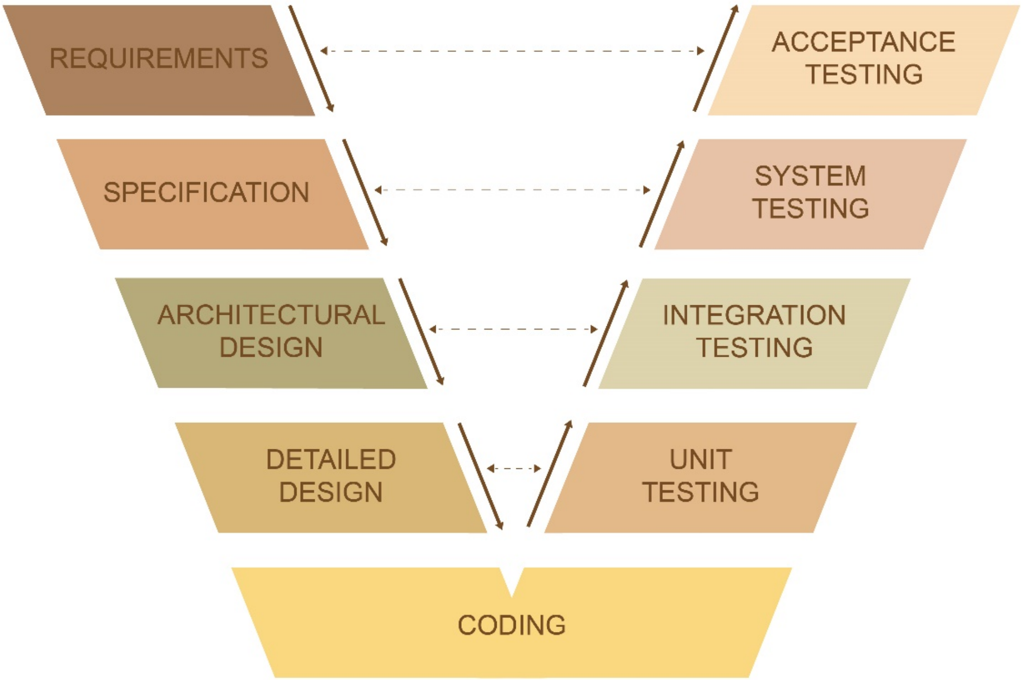


Figure 4. V-model (adapted from Marick 1999)

3.2.3 Spiral Model

The spiral model is a spiral (Figure 5) with a new product version on each turn. As soon as there is a new version, the project requirements are specified again, its quality is determined, and further work is planned for the next turn.

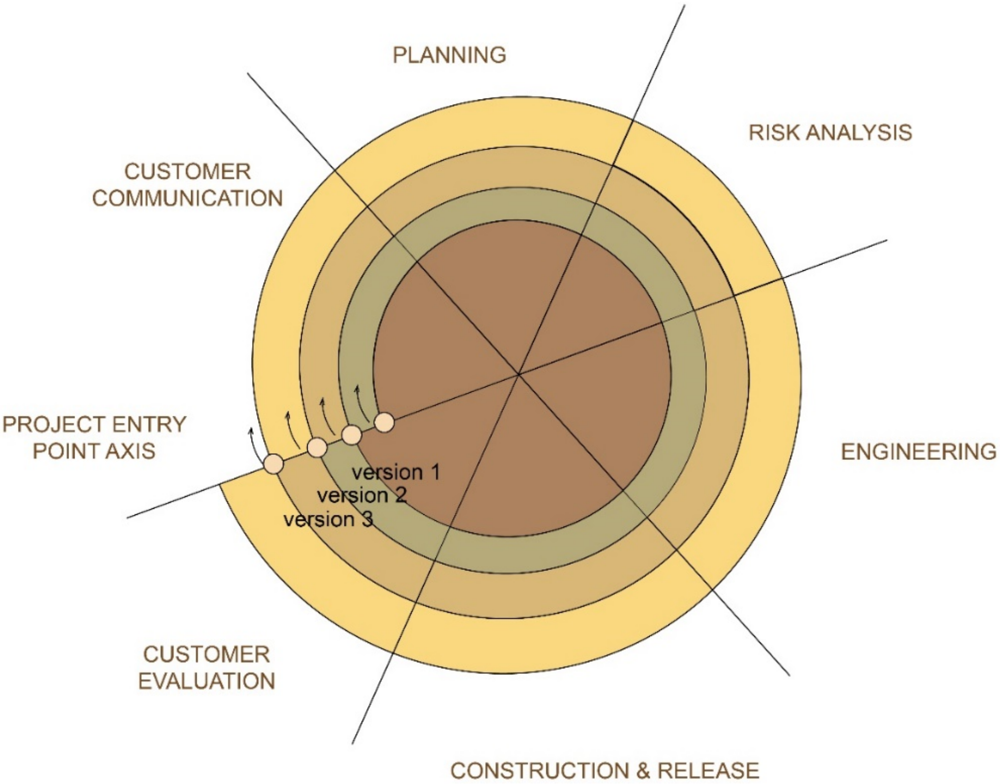


Figure 5. Spiral model (adapted from Boehm 2000)

The use of a spiral model allows the customer to see the system in the process of development and make adjustments to their requirements if necessary. The drawback of the model is the difficulty of determining the moment of transition between stages as well as the delay of project execution. (Boehm 2000, 3-4)

According to the results of the analysis, it was decided to use a spiral model as it gives more freedom to the developer and provides flexibility in interaction with potential customers.

Version control will be provided by a distributed file management system Git.

3.3 System Requirements

3.3.1 Structure and System Operation Requirements

The system under development shall be centralized, i.e. all data must be located in a central repository.

The system under development shall be an information structure available on the Internet under a certain domain name. (Software Engineering Tutorial 2020)

The system shall consist of interrelated sections with clearly separated functions.

The following sections are proposed to be highlighted in the system:

- User account management section – provides authentication and the ability to create a new account.
- News section – allows to view, change, delete, and create news.
- Lecture management section – allows to view, change, delete lecture materials.
- Course management section – allows to view, add, delete, modify courses.
- Group management section – allows teachers to create, modify, and delete groups.

3.3.2 List of Subsystems and Their Purpose

The following functional subsystems are proposed to be used in the system:

- User account management subsystem

  The user account management subsystem should provide the possibility of user authentication. Authentication should take place using a static login-password pair. The user password shall be stored in the database as a hash sum. In order to ensure the information security of a user, his or her password must not be stored in an open format. The subsystem also provides authorization and registration as well as stores primary information for the role-based subsystem.

- Role-based subsystem

  The role-based subsystem organizes the distribution of access rights between users. The access rights in the system are divided into two layers. The first layer contains information about the user role (student, teacher, administrator). The second layer regulates access of the first two groups to courses through the subscription system.

- Materials management subsystem

  The materials management subsystem performs the function of control over text and media materials: texts of lectures, news, and descriptions. The subsystem includes a WYSIWYG material editor that allows authorized users to apply HTML tags while writing texts. The subsystem allows to delete, change, add, edit materials.

- Group management subsystem

The group management subsystem provides the second level of the role system. Teachers can flexibly manage access to materials for different students using the group system. Access to groups is provided by means of secret phrases defined by the teacher.

The flexibility of the subsystem is ensured by the fact that the teacher is free to create the groups he or she needs. For example, a teacher may create a separate group for students who are behind schedule.

3.3.3 Compatibility Requirements

In order for users to work with the system, they shall use computers that provide the possibility to work over the Internet using a universal client application – a web browser. (Sheldon 1998, 5)

3.3.4 Operation Modes Requirements

The system shall support the following operation modes:

- Main operation mode – all subsystems operate in the normal mode.
- Restricted operation mode – a mode in which one or more subsystems are disabled for adjustment or modernization work.

3.3.5 Reliability Requirements

Reliability shall be ensured by means of:

- the use of technical means, system, and basic software corresponding to the class of tasks to be solved.
- compliance with the rules of operation and technical maintenance of software and hardware.
- preliminary training of users and maintenance personnel. (Reliability Basics 2007)

3.3.6 Ergonomic and Technical-Aesthetic Requirements

In terms of external design:

- The interfaces of the subsystems shall be consistent.
- The system interface shall be clear and user-friendly and shall not be overloaded with graphic elements.

- If an error occurs in the operation of the subsystem, a message shall be displayed on the screen with the error name and recommendations for its correction in English.
- The system shall support the English language.

3.3.7 Data Protection Requirements

The information security of the system shall meet the following requirements:

- The system shall be protected by a complex of software and hardware tools and organizational measures supporting them.
- System protection shall be provided at all technological stages of information processing and in all operation modes, including during repair and maintenance works.
- User passwords shall only be stored, transferred, and used as hash sums. (Datatilsynet 2017)

3.4 Data Structure Requirements

The application uses a large array of data, which is convenient to store as a relational database. The database shall contain the following objects (tables):

- **Users** – stores information about users registered in the system, their personal data, and data for verification of user authorization (Table 1). Also contains the user role in the system.

Table 1. Users table structure

Attribute	**Field type, restriction**
User ID	Int(11), AutoInc, !null
Username	Varchar(16) !null
Password hash sum	Varchar(255)
Surname	Varchar(42)
Name	Varchar(42)
Patronymic	Varchar(42)
Session hash sum	Varchar(32)
Last IP address	Varchar(16)

Role	Varchar(8), !null
User agent	Varchar(255)
Group	Varchar(12)

- **News** – contains news posted on the main page of the site (Table 2).

  Table 2. News table structure

Attribute	Field type, restriction
News ID	Int(11), AutoInc, !null
Title	Varchar(100), !null
News text	Longtext, !null
Importance tag	Tinyint(4), !null
Creation date	Timestamp, !null

- **Groups** – includes information about groups created by teachers and their code phrases (Table 3).

  Table 3. Groups table structure

Attribute	Field type, restriction
Group ID	Int(11), AutoInc, !null
Passcode	Char(16), !null
Expiry date	Timestamp
Teacher ID	Int(11)
Title	Varchar(16), !null

- **Disciplines** – contains information about courses created in the system (Table 4).

  Table 4. Disciplines table structure

Attribute	Field type, restriction
Discipline ID	Int(11), AutoInc, !null
Title	Varchar(64), !null
Description	Varchar(144)
Creator ID	Int(11)

- **Group_access** – contains information about the courses available to users belonging to certain groups (Table 5).

  Table 5. Group_access table structure

Attribute	**Field type, restriction**
Group ID	Int(11), !null
Discipline ID	Int(11), !null

- **Lectures** – contains information about the lectures of each course (Table 6).

  Table 6. Lectures table structure

Attribute	**Field type, restriction**
Lecture ID	Int(11), AutoInc, !null
Title	Varchar(80), !null
Description	Longtext, !null
Creation date	Timestamp
Update date	Timestamp, current on update
Discipline ID	Int(11)

- **Attachments** – contains data about attachments to lectures (Table 7).

  Table 7. Attachments table structure

Attribute	**Field type, restriction**
Attachment ID	Int(11), AutoInc, !null
Lecture ID	Int(11)
URL	Varchar(255)
Type	Varchar(5)
Title	Varchar(100), !null
Description	Varchar(255)
Creation date	Timestamp

- **Subscriptions** – contains information about which groups the user belongs to (Table 8).

Table 8. Subscriptions table structure

Attribute	Field type, restriction
User ID	Int(11), !null
Group ID	Int(11), !null
Editor's flag	Tinyint(1), !null

- **Group_progress** – contains information about the availability of materials for users of a particular group (Table 9).

Table 9. Group_progress table structure

Attribute	Field type, restriction
Group ID	Int(11), !null
Lecture ID	Int(11), !null
Deadline date	Timestamp
Visibility flag	Tinyint(1), !null

3.5 Subsystem Functional Characteristics Requirements

3.5.1 User Account Management Subsystem

The user account management subsystem shall provide the following functionalities:

- Automatic user authorization – shall be performed if the user's cookie contains a session hash that matches the one saved in the database. Also, for automatic authorization, the User Agent in the user's browser and IP address must match those in the database.
- Manual authorization – the system compares user data with database data using the login and password specified by the user.
- Logout – provides a secure end to the user session: deletes cookies in the browser and also deletes information for fast authorization from the database.
- Registration – provides an opportunity to create a new account for the user.

3.5.2 Role-Based Subsystem

The role-based subsystem shall provide the following functionalities:

- Comparison of user role with available functionalities of other subsystems

- Provision of access to permitted functions.

3.5.3 Materials Management Subsystem

Materials management subsystem regulates the work with news, lectures, and attachments, providing them with the following set of features:

- Addition of news/lectures/attachments
- Modification of news/lectures/attachments
- Deletion of news/lectures/attachments
- Provision of access to news/lectures/attachments according to the permissions assigned by the role-based system.

4 DATABASE AND APPLICATION DEVELOPMENT

4.1 System Appearance and Features

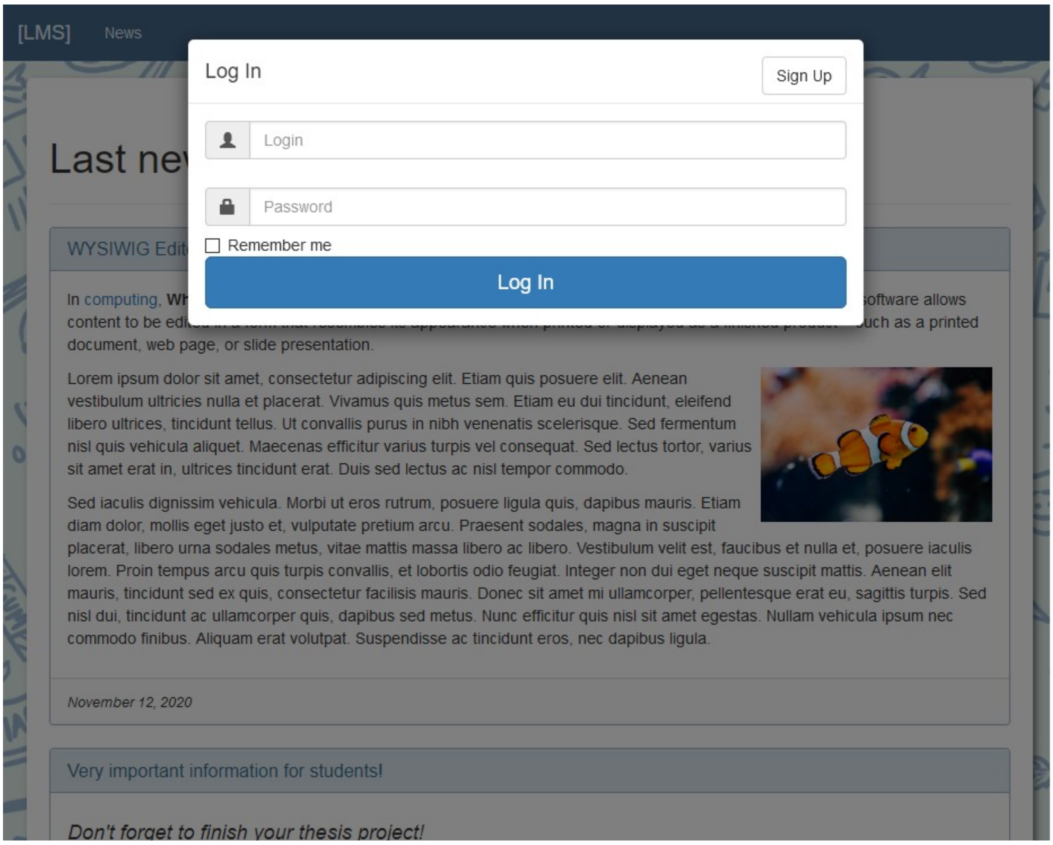
4.1.1 Registration and Authorization

Work in the system is possible only after authorization, therefore when logging on to the site unauthorized users will see a modal window offering to login or register (Image 1). This window is displayed over the main content of the site and cannot be skipped. Standard login-password pair is used for login authorization.

In the upper right corner of the window, there is the Sign Up button, which switches the window into the registration mode (Image 2). To register in the system, a user must provide the following information about themselves:

- Login and password that will be used for authorization.
- First and last name
- Name of the study group
- Passphrase

In order to exclude unwanted registrations in the system, it is necessary to enter a passcode during registration.



Last ne

WYSIWYG Edit

In computing, Wh … software allows content to be edi … uch as a printed document, web page, or slide presentation.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam quis posuere elit. Aenean vestibulum ultricies nulla et placerat. Vivamus quis metus sem. Etiam eu dui tincidunt, eleifend libero ultrices, tincidunt tellus. Ut convallis purus in nibh venenatis scelerisque. Sed fermentum nisl quis vehicula aliquet. Maecenas efficitur varius turpis vel consequat. Sed lectus tortor, varius sit amet erat in, ultrices tincidunt erat. Duis sed lectus ac nisl tempor commodo.

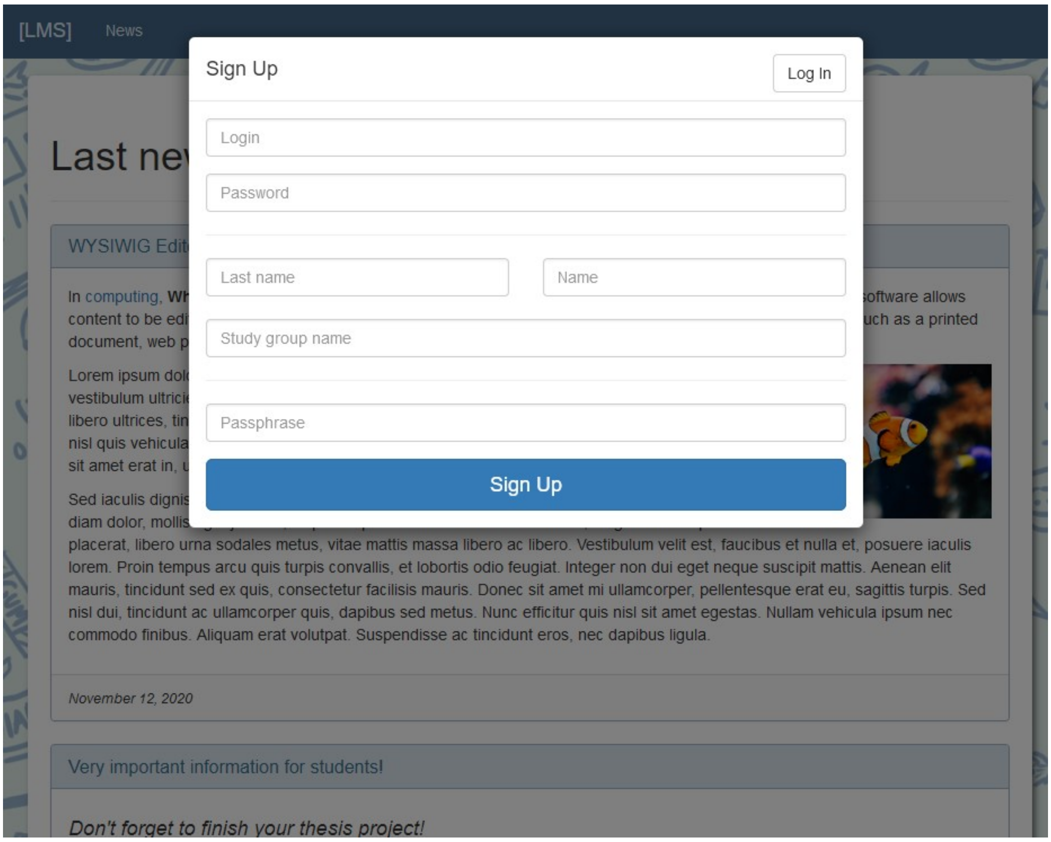
Sed iaculis dignissim vehicula. Morbi ut eros rutrum, posuere ligula quis, dapibus mauris. Etiam diam dolor, mollis eget justo et, vulputate pretium arcu. Praesent sodales, magna in suscipit placerat, libero urna sodales metus, vitae mattis massa libero ac libero. Vestibulum velit est, faucibus et nulla et, posuere iaculis lorem. Proin tempus arcu quis turpis convallis, et lobortis odio feugiat. Integer non dui eget neque suscipit mattis. Aenean elit mauris, tincidunt sed ex quis, consectetur facilisis mauris. Donec sit amet mi ullamcorper, pellentesque erat eu, sagittis turpis. Sed nisl dui, tincidunt ac ullamcorper quis, dapibus sed metus. Nunc efficitur quis nisl sit amet egestas. Nullam vehicula ipsum nec commodo finibus. Aliquam erat volutpat. Suspendisse ac tincidunt eros, nec dapibus ligula.

November 12, 2020

Very important information for students!

Don't forget to finish your thesis project!

Image 1. Modal login window



Last ne

WYSIWYG Edit

In computing, Wh … software allows content to be edi … uch as a printed document, web p

Lorem ipsum dol vestibulum ultrici libero ultrices, tin nisl quis vehicula sit amet erat in, u

Sed iaculis dignis diam dolor, mollis … placerat, libero urna sodales metus, vitae mattis massa libero ac libero. Vestibulum velit est, faucibus et nulla et, posuere iaculis lorem. Proin tempus arcu quis turpis convallis, et lobortis odio feugiat. Integer non dui eget neque suscipit mattis. Aenean elit mauris, tincidunt sed ex quis, consectetur facilisis mauris. Donec sit amet mi ullamcorper, pellentesque erat eu, sagittis turpis. Sed nisl dui, tincidunt ac ullamcorper quis, dapibus sed metus. Nunc efficitur quis nisl sit amet egestas. Nullam vehicula ipsum nec commodo finibus. Aliquam erat volutpat. Suspendisse ac tincidunt eros, nec dapibus ligula.

November 12, 2020

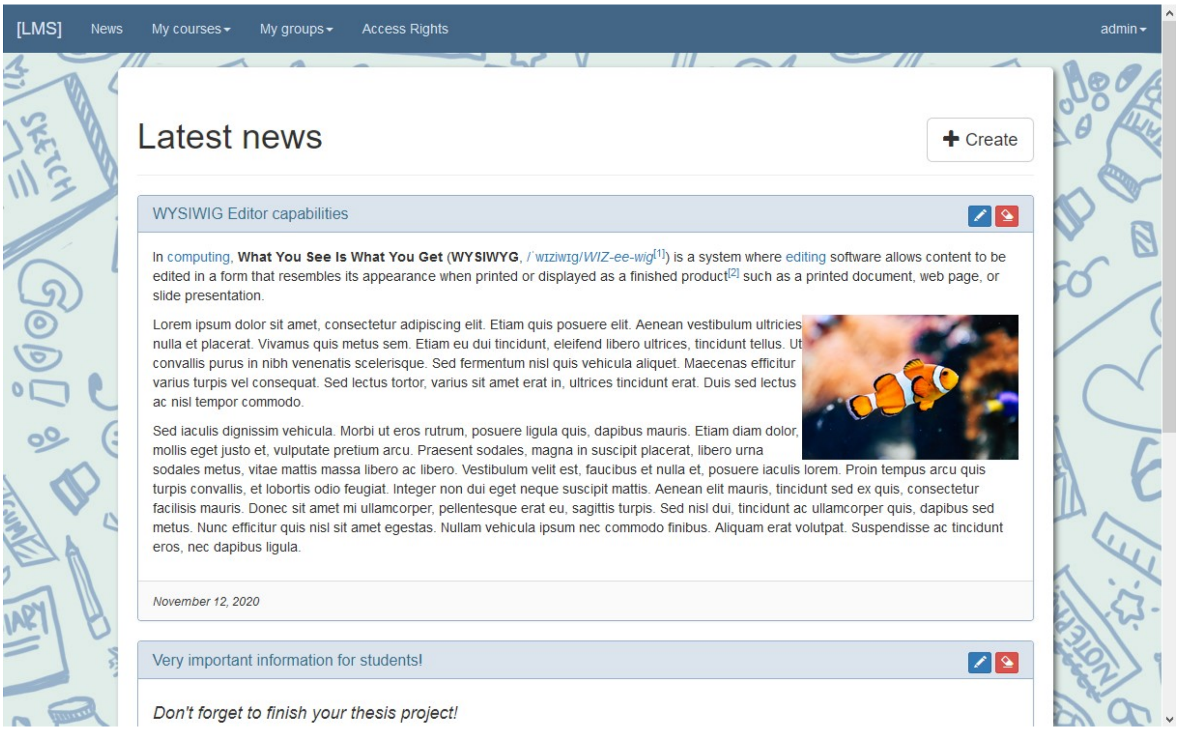
Very important information for students!

Don't forget to finish your thesis project!

Image 2. Modal sign up window

4.1.2 News Section

The news page is the main page of the site. All system users can view the news, and users with administrator rights can edit, add and delete news. Image 3 shows the news page in the administrator view. For ordinary users, it differs only in the absence of some controls.



Latest news

WYSIWIG Editor capabilities

In computing, **What You See Is What You Get** (**WYSIWYG**, *ˈwɪziwɪɡ*/*WIZ-ee-wig*[1]) is a system where editing software allows content to be edited in a form that resembles its appearance when printed or displayed as a finished product[2] such as a printed document, web page, or slide presentation.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam quis posuere elit. Aenean vestibulum ultricies nulla et placerat. Vivamus quis metus sem. Etiam eu dui tincidunt, eleifend libero ultrices, tincidunt tellus. Ut convallis purus in nibh venenatis scelerisque. Sed fermentum nisl quis vehicula aliquet. Maecenas efficitur varius turpis vel consequat. Sed lectus tortor, varius sit amet erat in, ultrices tincidunt erat. Duis sed lectus ac nisl tempor commodo.

Sed iaculis dignissim vehicula. Morbi ut eros rutrum, posuere ligula quis, dapibus mauris. Etiam diam dolor, mollis eget justo et, vulputate pretium arcu. Praesent sodales, magna in suscipit placerat, libero urna sodales metus, vitae mattis massa libero ac libero. Vestibulum velit est, faucibus et nulla et, posuere iaculis lorem. Proin tempus arcu quis turpis convallis, et lobortis odio feugiat. Integer non dui eget neque suscipit mattis. Aenean elit mauris, tincidunt sed ex quis, consectetur facilisis mauris. Donec sit amet mi ullamcorper, pellentesque erat eu, sagittis turpis. Sed nisl dui, tincidunt ac ullamcorper quis, dapibus sed metus. Nunc efficitur quis nisl sit amet egestas. Nullam vehicula ipsum nec commodo finibus. Aliquam erat volutpat. Suspendisse ac tincidunt eros, nec dapibus ligula.

November 12, 2020

Very important information for students!

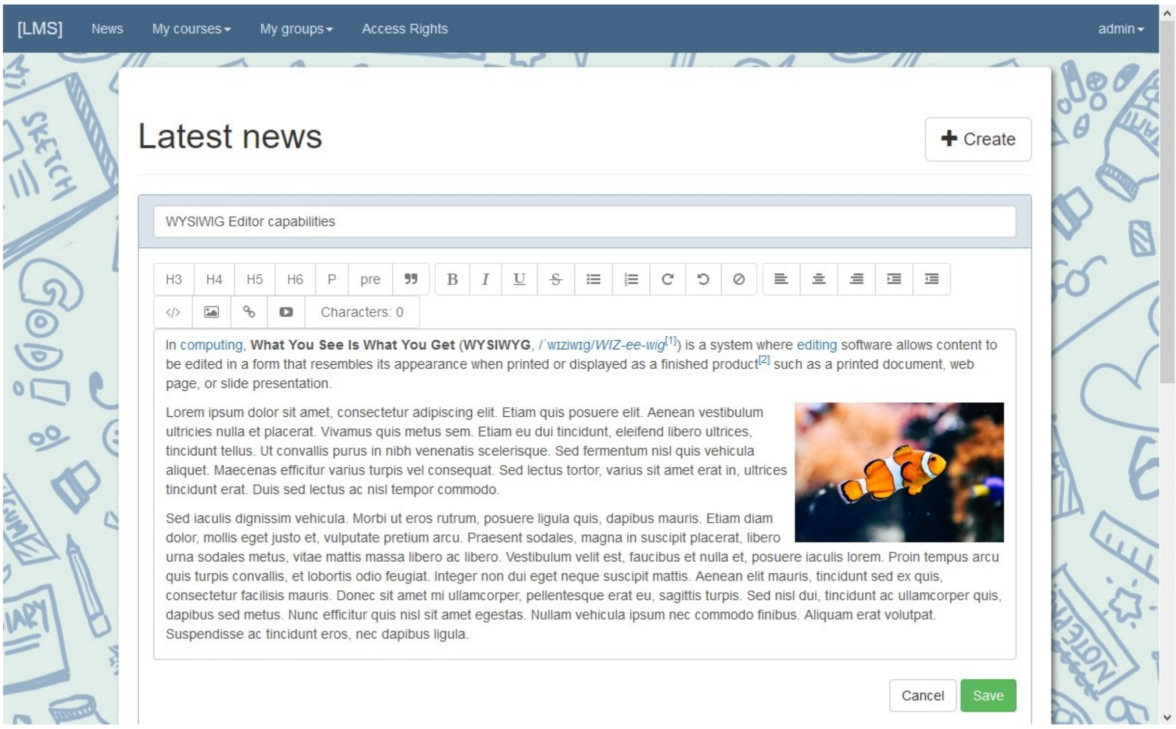
Don't forget to finish your thesis project!

Image 3. News page in the administrator view

The following controls are present on the page:

- Newer button and Older button at the bottom of the page – switches pages with news. By default, there are three news on the page. Available to all users.
- Create button at the top of the page – creates a new news template. Available only to administrator.
- Edit button (pencil symbol) – switches the news to edit mode. Available only to the administrator.
- Delete button (eraser symbol) – deletes news after operation confirmation. Available only to administrator.

It is also allowed to use HTML features for news formatting. In order to simplify news design, the system uses a full-fledged WYSIWYG editor (Image 4).



Latest news

WYSIWIG Editor capabilities

In computing, **What You See Is What You Get** (**WYSIWYG**, /ˈwɪziwɪɡ/*WIZ-ee-wig*[1]) is a system where editing software allows content to be edited in a form that resembles its appearance when printed or displayed as a finished product[2] such as a printed document, web page, or slide presentation.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam quis posuere elit. Aenean vestibulum ultricies nulla et placerat. Vivamus quis metus sem. Etiam eu dui tincidunt, eleifend libero ultrices, tincidunt tellus. Ut convallis purus in nibh venenatis scelerisque. Sed fermentum nisl quis vehicula aliquet. Maecenas efficitur varius turpis vel consequat. Sed lectus tortor, varius sit amet erat in, ultrices tincidunt erat. Duis sed lectus ac nisl tempor commodo.

Sed iaculis dignissim vehicula. Morbi ut eros rutrum, posuere ligula quis, dapibus mauris. Etiam diam dolor, mollis eget justo et, vulputate pretium arcu. Praesent sodales, magna in suscipit placerat, libero urna sodales metus, vitae mattis massa libero ac libero. Vestibulum velit est, faucibus et nulla et, posuere iaculis lorem. Proin tempus arcu quis turpis convallis, et lobortis odio feugiat. Integer non dui eget neque suscipit mattis. Aenean elit mauris, tincidunt sed ex quis, consectetur facilisis mauris. Donec sit amet mi ullamcorper, pellentesque erat eu, sagittis turpis. Sed nisl dui, tincidunt ac ullamcorper quis, dapibus sed metus. Nunc efficitur quis nisl sit amet egestas. Nullam vehicula ipsum nec commodo finibus. Aliquam erat volutpat. Suspendisse ac tincidunt eros, nec dapibus ligula.

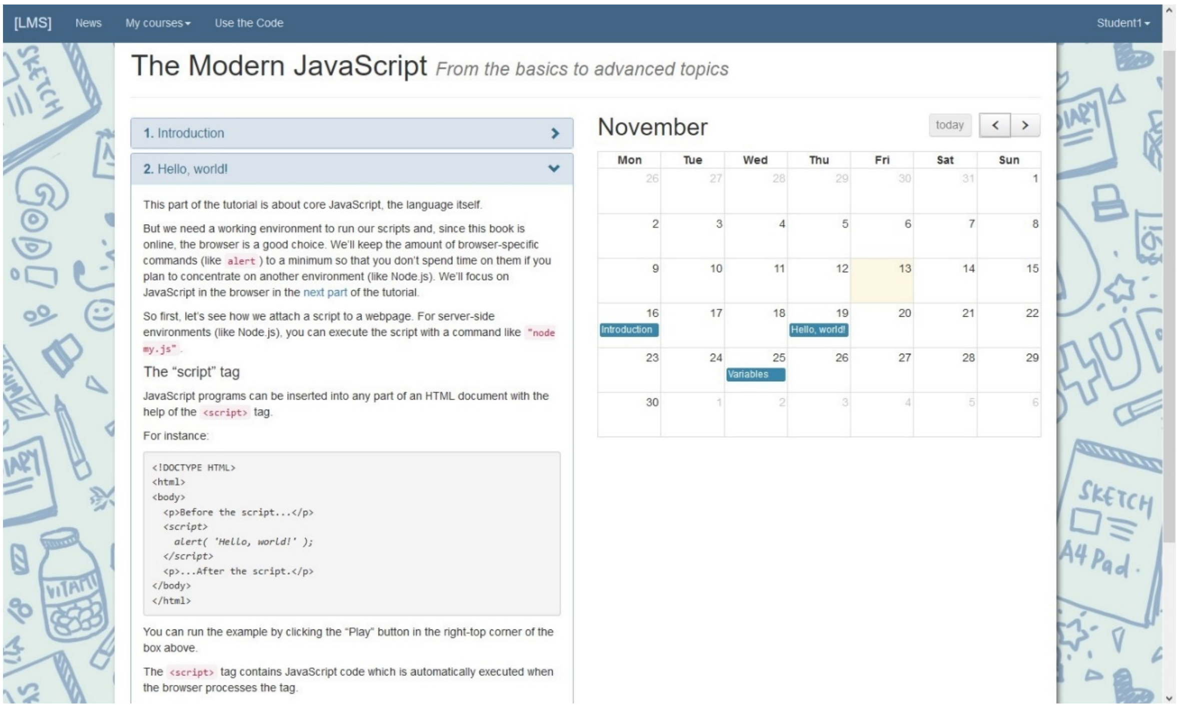
Image 4. News in edit mode

4.1.3 My Courses Section

The My Courses section contains a list of courses available to the user. This list is generated in different ways, depending on the user's role in the system:

- Administrators – can see all courses placed in the system.
- Teachers – can see their courses and those available by subscription.
- Students – can see the courses available to them by subscription.

The course page (Image 5) contains a list of lectures and a calendar with information about assignment deadlines. In the teacher view, the calendar will not be displayed and there will be additional controls like those in the News section.



The Modern JavaScript *From the basics to advanced topics*

This part of the tutorial is about core JavaScript, the language itself.

But we need a working environment to run our scripts and, since this book is online, the browser is a good choice. We'll keep the amount of browser-specific commands (like `alert`) to a minimum so that you don't spend time on them if you plan to concentrate on another environment (like Node.js). We'll focus on JavaScript in the browser in the next part of the tutorial.

So first, let's see how we attach a script to a webpage. For server-side environments (like Node.js), you can execute the script with a command like `"node my.js"`.

The "script" tag

JavaScript programs can be inserted into any part of an HTML document with the help of the `<script>` tag.

For instance:

```
<!DOCTYPE HTML>
<html>
<body>
  <p>Before the script...</p>
  <script>
    alert( 'Hello, world!' );
  </script>
  <p>...After the script.</p>
</body>
</html>
```

You can run the example by clicking the "Play" button in the right-top corner of the box above.

The `<script>` tag contains JavaScript code which is automatically executed when the browser processes the tag.

November

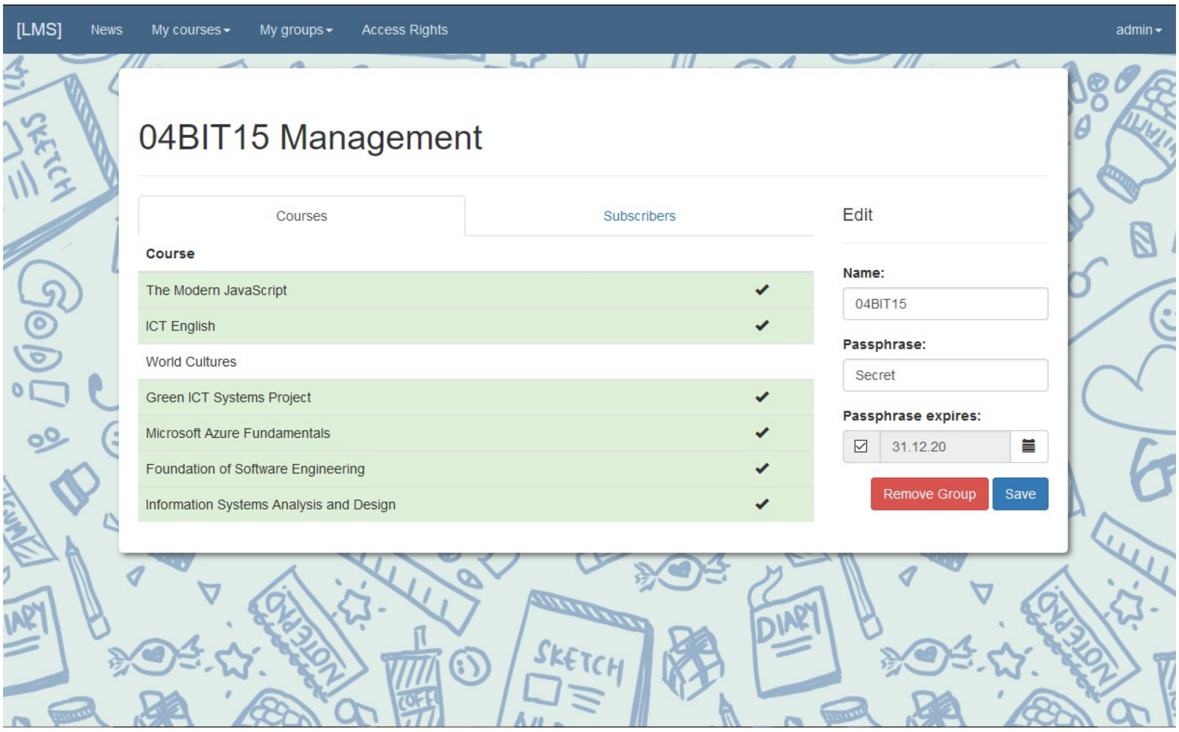
Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16 Introduction	17	18	19 Hello, world!	20	21	22
23	24	25 Variables	26	27	28	29
30	1	2	3	4	5	6

Image 5. Course page in the student view

4.1.4 My Groups Section

My Groups section is only available to teachers and administrators, and it is intended for management of groups.

The group page is divided into two tabs: Courses (Image 6) and Subscribers (Image 7). On the courses tab, the teacher can specify which courses the group subscribers should see. Changes are made by selecting the required courses from the table.



04BIT15 Management

Course	
The Modern JavaScript	✔
ICT English	✔
World Cultures	
Green ICT Systems Project	✔
Microsoft Azure Fundamentals	✔
Foundation of Software Engineering	✔
Information Systems Analysis and Design	✔

Edit

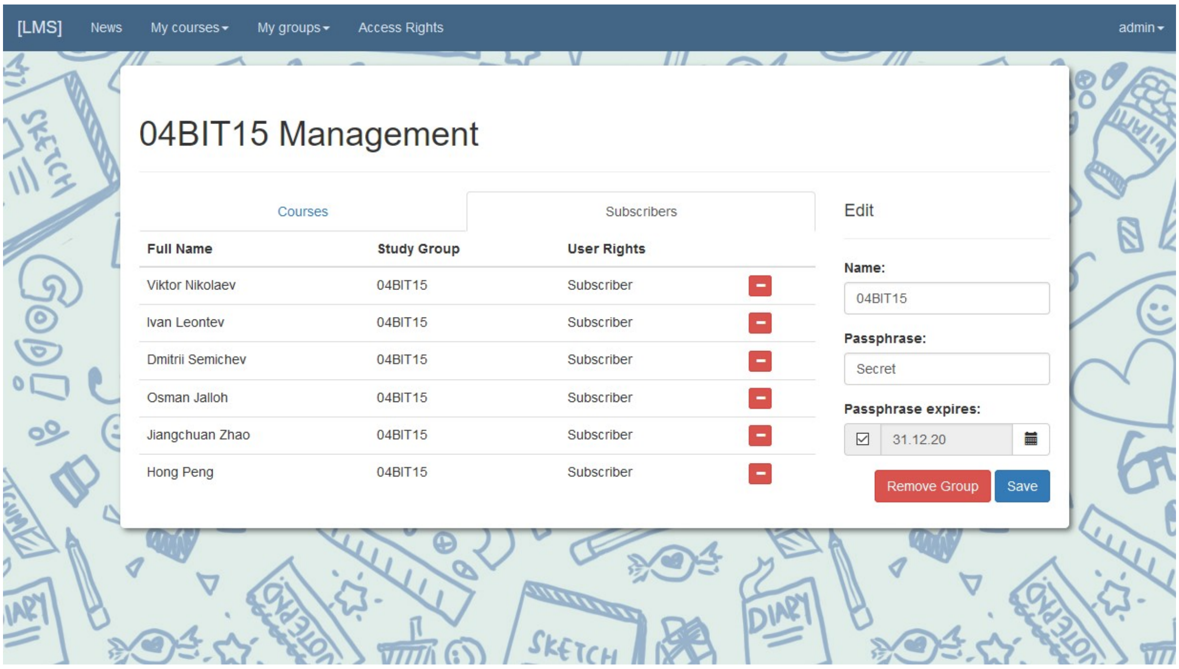
Name: 04BIT15

Passphrase: Secret

Passphrase expires: ☑ 31.12.20

Image 6. Courses tab of My groups section

On the Subscribers tab, a teacher can see a list of subscribers to the group. The teacher is not free to add students to the group, however, he or she can remove them at his or her discretion. For this purpose, there is a button with a "minus" symbol next to the name of each subscriber. Besides, on this page the teacher can assign editors of the group. Editors can modify the course lecture materials, but they cannot make changes to the group.



04BIT15 Management

Full Name	Study Group	User Rights	
Viktor Nikolaev	04BIT15	Subscriber	−
Ivan Leontev	04BIT15	Subscriber	−
Dmitrii Semichev	04BIT15	Subscriber	−
Osman Jalloh	04BIT15	Subscriber	−
Jiangchuan Zhao	04BIT15	Subscriber	−
Hong Peng	04BIT15	Subscriber	−

Edit

Name: 04BIT15

Passphrase: Secret

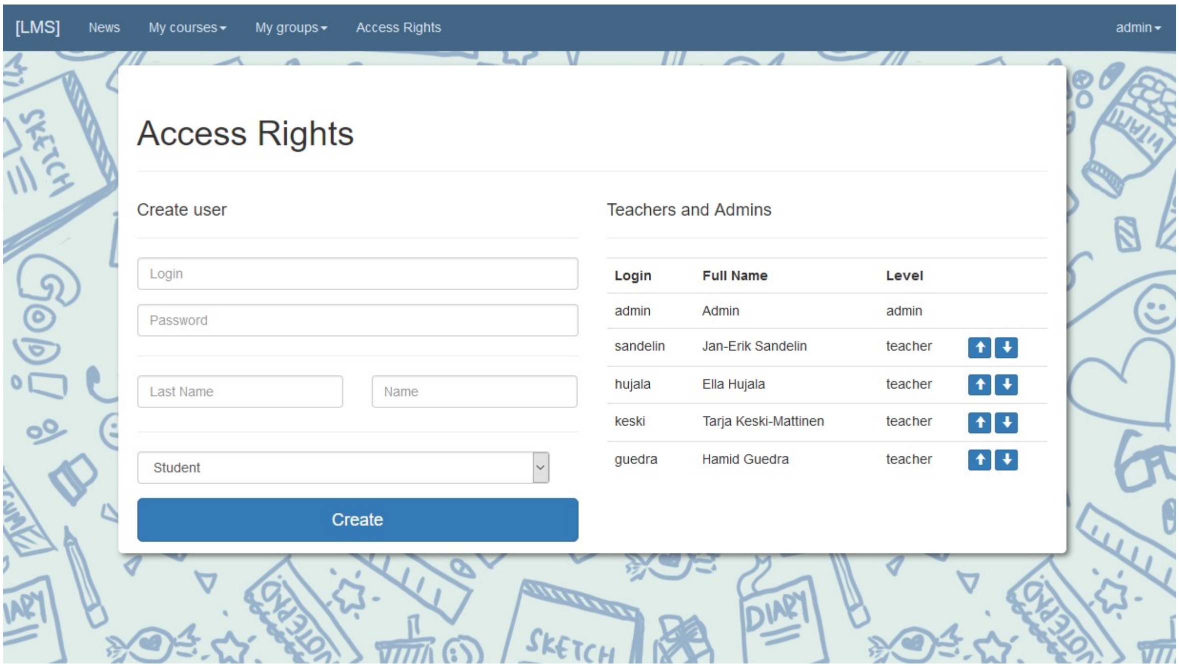
Passphrase expires: ☑ 31.12.20

Image 7. Subscribers tab of My groups section

On the right side of the group page, regardless of the tab selected, there is a group editing block. The teacher can change the group name, the passphrase, and its expiration date from there. It is allowed to use a passcode for an unlimited period of time; for this purpose, it is necessary to uncheck the checkbox in the Passphrase expires field. The passcode is required for students to subscribe to the group. Using the Remove Group button, the teacher can completely remove the group and associated data from the system.

4.1.5 Access Rights Section

The access rights section (Image 8) is available only to system administrators. With the help of this section, an administrator can create a new user with the necessary level of rights (Student, Teacher, Administrator) and manage roles of existing administrators and teachers in the system.



Access Rights

Create user

Teachers and Admins

Login	Full Name	Level	
admin	Admin	admin	
sandelin	Jan-Erik Sandelin	teacher	↑ ↓
hujala	Ella Hujala	teacher	↑ ↓
keski	Tarja Keski-Mattinen	teacher	↑ ↓
guedra	Hamid Guedra	teacher	↑ ↓

Image 8. Access rights section

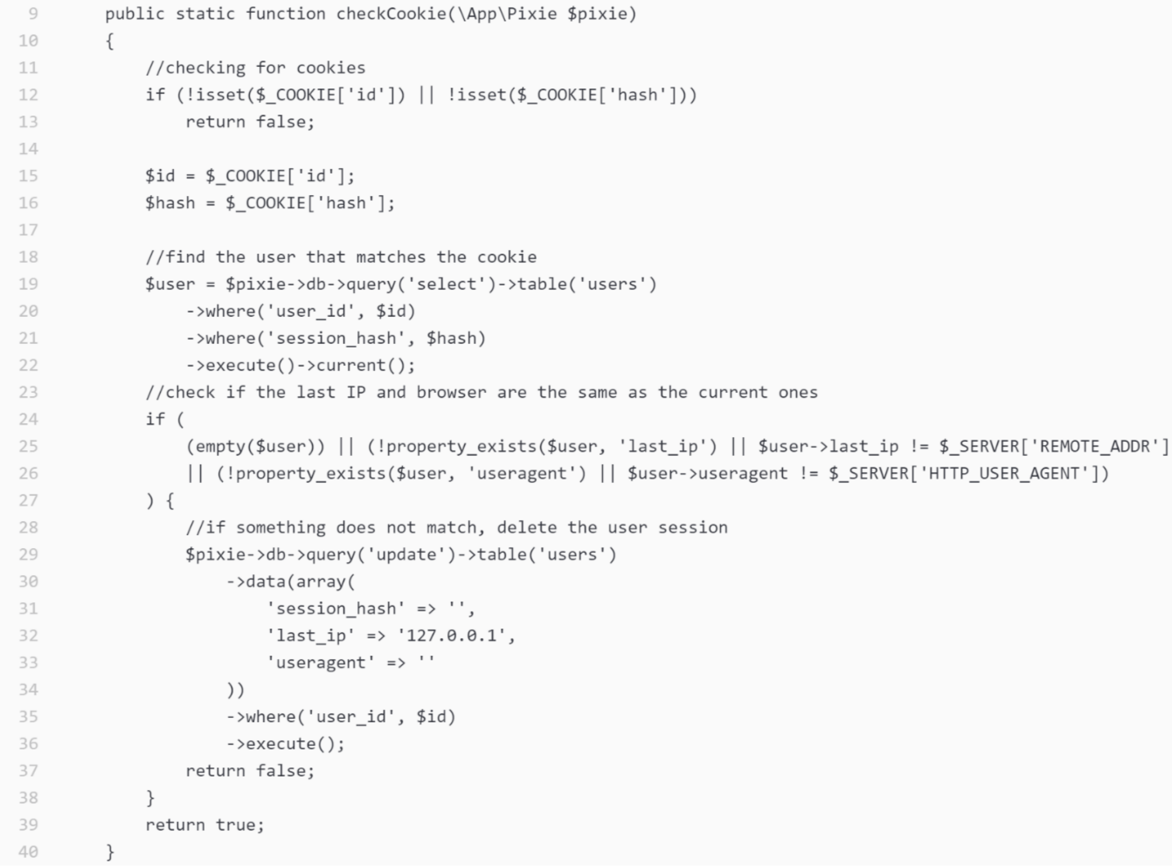
The page is visually divided into two blocks: a simplified registration form with the option to choose the user role on the left and a table with a list of teachers and administrators on the right. In order to change the role of an existing user, the administrator can use the up or down buttons to increase or decrease the user's rights.

4.2 Functionality Implementation

4.2.1 Authorization

In order to implement authorization and further support of the session, a whole range of interrelated system elements is involved. In general, the authorization process after the start of the application is as follows:

1. At initialization of the Angular application, the API checkCookie function is called (Image 9). The function checks if the session can be restored with cookies saved in the user's browser.



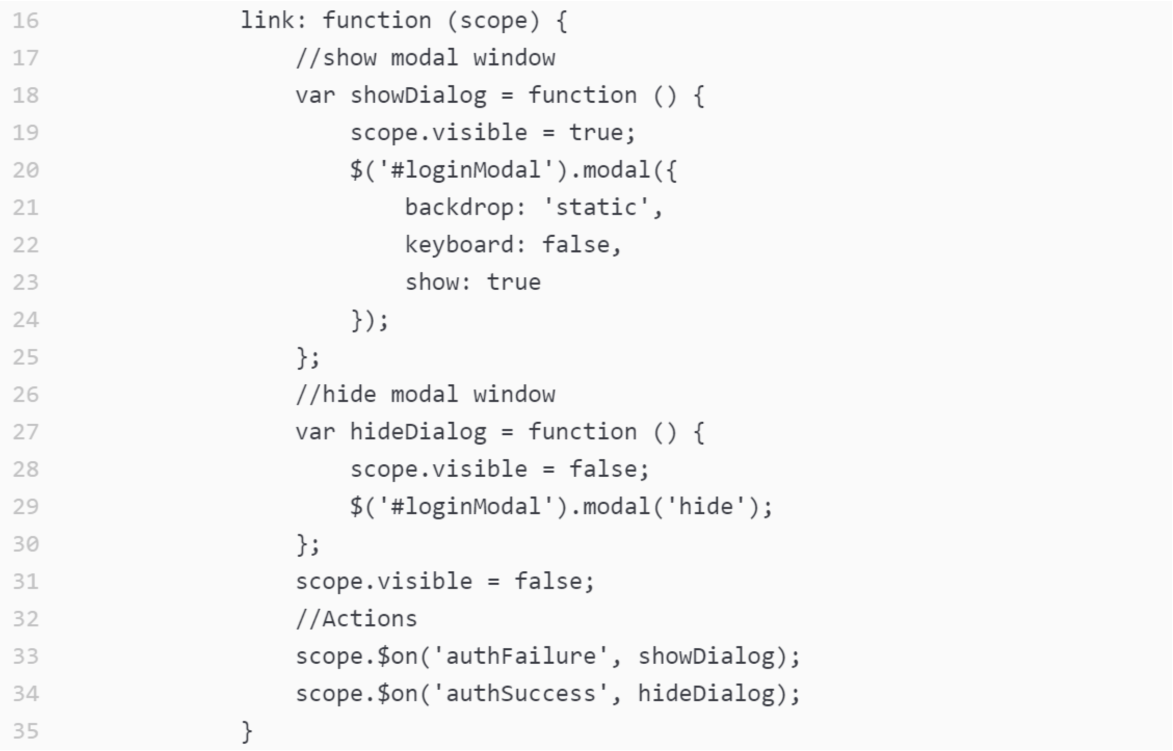
```
public static function checkCookie(\App\Pixie $pixie)
{
    //checking for cookies
    if (!isset($_COOKIE['id']) || !isset($_COOKIE['hash']))
        return false;

    $id = $_COOKIE['id'];
    $hash = $_COOKIE['hash'];

    //find the user that matches the cookie
    $user = $pixie->db->query('select')->table('users')
        ->where('user_id', $id)
        ->where('session_hash', $hash)
        ->execute()->current();
    //check if the last IP and browser are the same as the current ones
    if (
        (empty($user)) || (!property_exists($user, 'last_ip') || $user->last_ip != $_SERVER['REMOTE_ADDR']
        || (!property_exists($user, 'useragent') || $user->useragent != $_SERVER['HTTP_USER_AGENT'])
    ) {
        //if something does not match, delete the user session
        $pixie->db->query('update')->table('users')
            ->data(array(
                'session_hash' => '',
                'last_ip' => '127.0.0.1',
                'useragent' => ''
            ))
            ->where('user_id', $id)
            ->execute();
        return false;
    }
    return true;
}
```

Image 9. Listing of checkCookie function

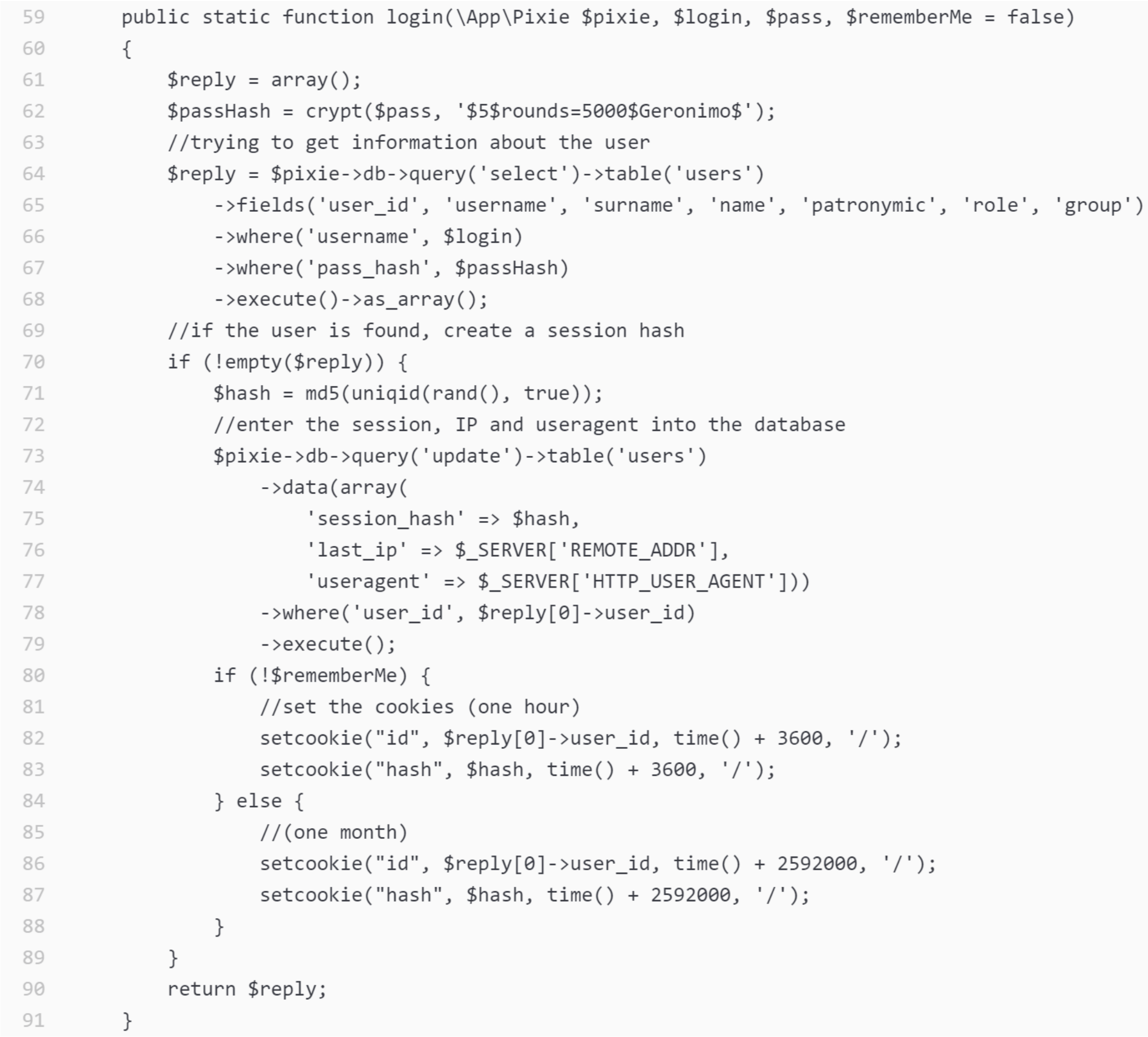
2. The Angular application notifies all child elements that authorization is not completed. The directive of the modal authorization window (Image 10) receives a message from the application and displays the modal login window. Control is transferred to the modal window controller.



```
            link: function (scope) {
                //show modal window
                var showDialog = function () {
                    scope.visible = true;
                    $('#loginModal').modal({
                        backdrop: 'static',
                        keyboard: false,
                        show: true
                    });
                };
                //hide modal window
                var hideDialog = function () {
                    scope.visible = false;
                    $('#loginModal').modal('hide');
                };
                scope.visible = false;
                //Actions
                scope.$on('authFailure', showDialog);
                scope.$on('authSuccess', hideDialog);
            }
```

Image 10. Listing of modal window directive functions

3. Once the data have been entered by the user, the API Auth function is called, which performs authorization using the data entered in the form (Image 11). During the function execution the password entered by the user is hashed by crypt function using SHA-1 algorithm. To ensure security, passwords are stored in the database as hashes. During authorization, a session hash is also created using the MD-5 algorithm. It is entered into the database and user cookies and allows to restore the user session.

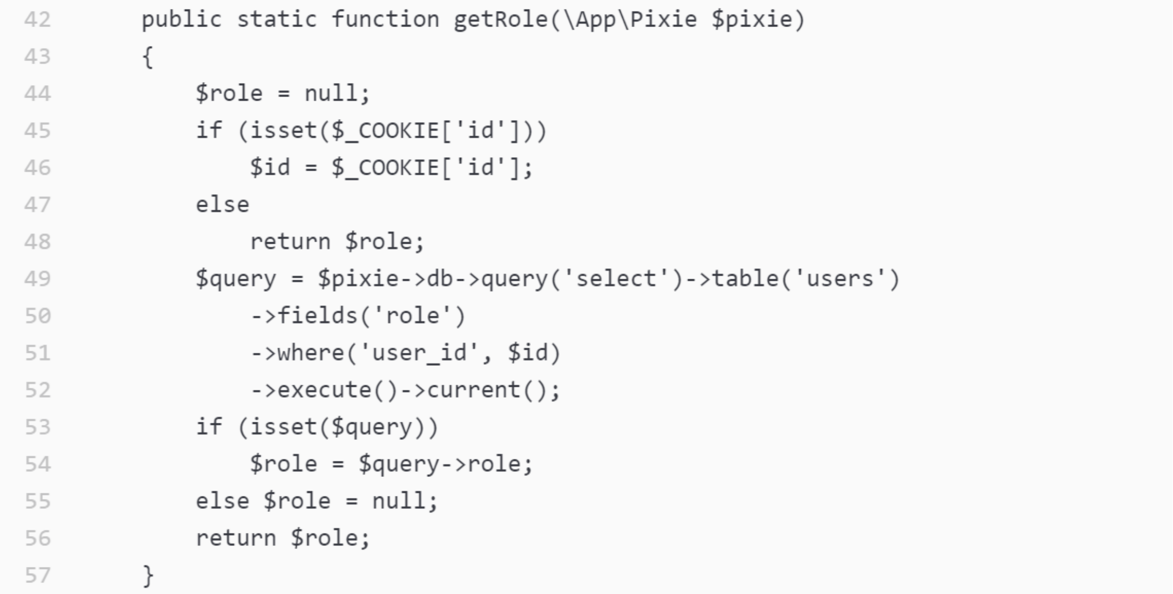


```
    public static function login(\App\Pixie $pixie, $login, $pass, $rememberMe = false)
    {
        $reply = array();
        $passHash = crypt($pass, '$5$rounds=5000$Geronimo$');
        //trying to get information about the user
        $reply = $pixie->db->query('select')->table('users')
            ->fields('user_id', 'username', 'surname', 'name', 'patronymic', 'role', 'group')
            ->where('username', $login)
            ->where('pass_hash', $passHash)
            ->execute()->as_array();
        //if the user is found, create a session hash
        if (!empty($reply)) {
            $hash = md5(uniqid(rand(), true));
            //enter the session, IP and useragent into the database
            $pixie->db->query('update')->table('users')
                ->data(array(
                    'session_hash' => $hash,
                    'last_ip' => $_SERVER['REMOTE_ADDR'],
                    'useragent' => $_SERVER['HTTP_USER_AGENT']))
                ->where('user_id', $reply[0]->user_id)
                ->execute();
            if (!$rememberMe) {
                //set the cookies (one hour)
                setcookie("id", $reply[0]->user_id, time() + 3600, '/');
                setcookie("hash", $hash, time() + 3600, '/');
            } else {
                //(one month)
                setcookie("id", $reply[0]->user_id, time() + 2592000, '/');
                setcookie("hash", $hash, time() + 2592000, '/');
            }
        }
        return $reply;
    }
```

Image 11. Listing of login function

4. If successful, the authorization service notifies the application modules of the completion of authorization, and information about the user is entered into the Session service for further use. Modal login window is removed from the screen.

5. Since the Angular application is written in JavaScript and runs on the user side and can be modified, there are a number of checks on the API side that are performed on each client request. Among them: login check, role check (Image 12), access rights check and some others.



```php
    public static function getRole(\App\Pixie $pixie)
    {
        $role = null;
        if (isset($_COOKIE['id']))
            $id = $_COOKIE['id'];
        else
            return $role;
        $query = $pixie->db->query('select')->table('users')
            ->fields('role')
            ->where('user_id', $id)
            ->execute()->current();
        if (isset($query))
            $role = $query->role;
        else $role = null;
        return $role;
    }
```

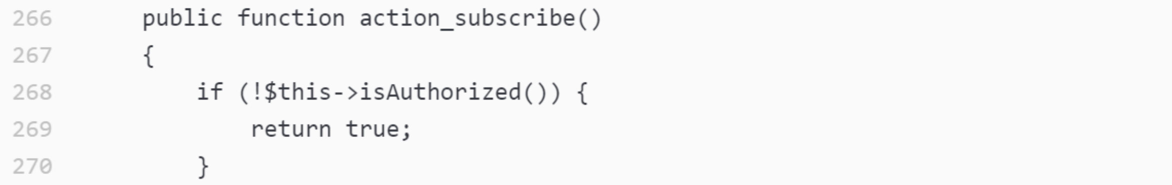
Image 12. Listing of getRole function

4.2.2 Subscription

On the example of subscription to one of the groups using a passphrase it is possible to see how the entire API works. The action_subscribe function is called when the user or front-end tries to access the /api/groups/subscribe page. At the input, the function requires a passcode, if the passcode is not passed through a query, the function stops executing and an error is returned.

There are a number of checks to be performed before adding a user to a group. The function performs the following checks:

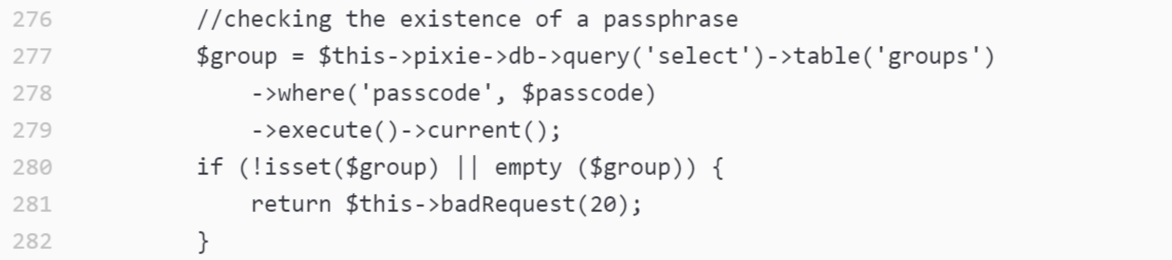
1. The authorization of the current user is checked (Image 13).



```php
    public function action_subscribe()
    {
        if (!$this->isAuthorized()) {
            return true;
        }
```

Image 13. User authorization check

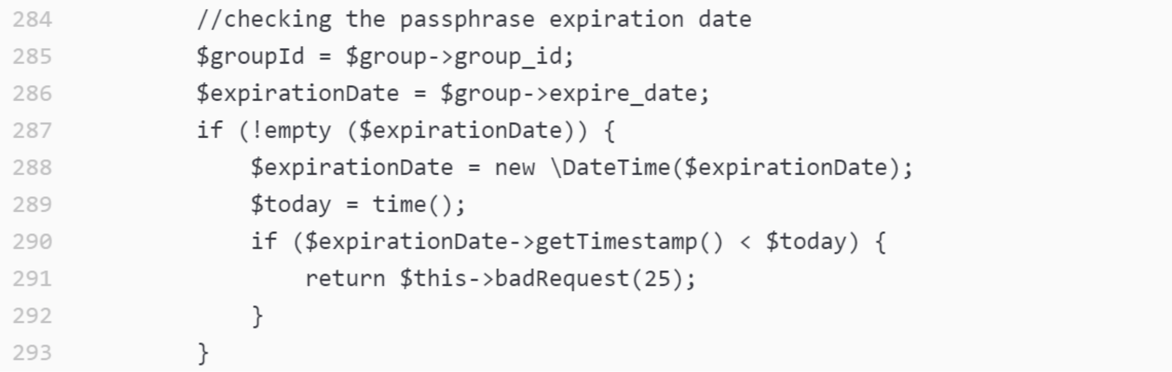
2. Checking the existence of a passcode in the database and its related group (Image 14).



```
        //checking the existence of a passphrase
        $group = $this->pixie->db->query('select')->table('groups')
            ->where('passcode', $passcode)
            ->execute()->current();
        if (!isset($group) || empty ($group)) {
            return $this->badRequest(20);
        }
```

Image 14. Checking the existence of a passphrase

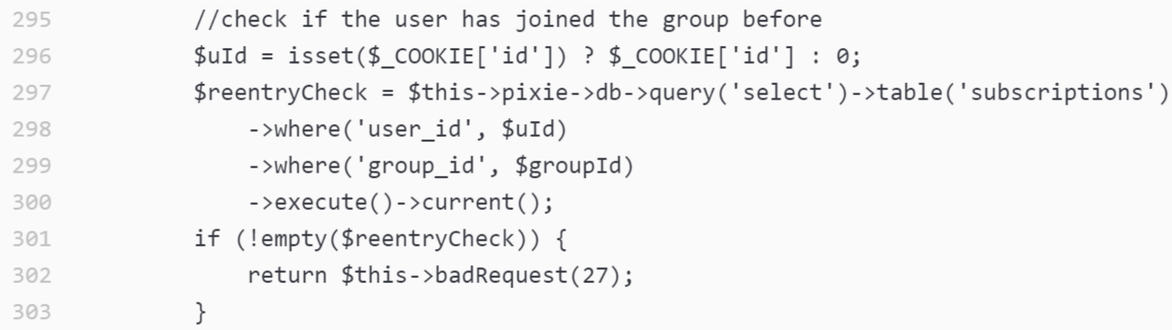
3. The expiration date of the passcode is compared to the current date (Image 15).



```
        //checking the passphrase expiration date
        $groupId = $group->group_id;
        $expirationDate = $group->expire_date;
        if (!empty ($expirationDate)) {
            $expirationDate = new \DateTime($expirationDate);
            $today = time();
            if ($expirationDate->getTimestamp() < $today) {
                return $this->badRequest(25);
            }
        }
```

Image 15. Checking the passphrase expiration date

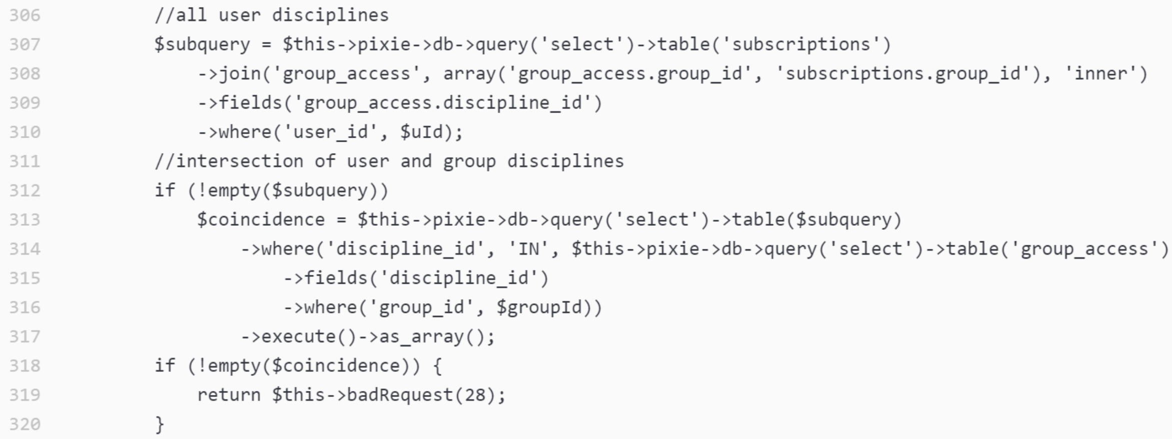
4. The list of user groups is checked (in case the user tries to join the same group again) (Image 16).



```
        //check if the user has joined the group before
        $uId = isset($_COOKIE['id']) ? $_COOKIE['id'] : 0;
        $reentryCheck = $this->pixie->db->query('select')->table('subscriptions')
            ->where('user_id', $uId)
            ->where('group_id', $groupId)
            ->execute()->current();
        if (!empty($reentryCheck)) {
            return $this->badRequest(27);
        }
```

Image 16. User groups check

5. Finally, the disciplines that the group provides access to and the disciplines that the user has access to are checked. There should be no coincidences. The check code is shown in Image 17.



```
        //all user disciplines
        $subquery = $this->pixie->db->query('select')->table('subscriptions')
            ->join('group_access', array('group_access.group_id', 'subscriptions.group_id'), 'inner')
            ->fields('group_access.discipline_id')
            ->where('user_id', $uId);
        //intersection of user and group disciplines
        if (!empty($subquery))
            $coincidence = $this->pixie->db->query('select')->table($subquery)
                ->where('discipline_id', 'IN', $this->pixie->db->query('select')->table('group_access')
                    ->fields('discipline_id')
                    ->where('group_id', $groupId))
                ->execute()->as_array();
        if (!empty($coincidence)) {
            return $this->badRequest(28);
        }
```

Image 17. Checking the intersection of user and group disciplines

5 CONCLUSIONS

5.1 Answers to the Research Questions

The main objective of this research was to explore the process of developing a learning management system based on a single page application architecture. In order to achieve this goal, the following answers to the research questions are provided below.

What is the architecture of the application under development?

It was initially assumed that the project will have extensive functionality, that is why the application uses MVC pattern. It makes the application flexible and well-structured: the presentation logic and application logic are separated from each other, which simplifies testing, debugging, and maintenance of the application, as well as further modification and code expansion.

Which tools and technologies to use to build the system?

The front-end is responsible for the client part of the project, application interface, and ways of interaction with it. Markup of web pages of the application is written in HTML language, and the appearance is described in CSS language. Twitter Bootstrap and AngularJS framework are also used in the project.

The back-end is the code responsible for application business logic and data handling. PHP is in charge of the application logic and MySQL is used as a DBMS. Besides that, PHPixie framework is also used in the project.

What are the existing solutions on the market?

At present, there are many solutions for creating learning management systems, but in this paper, two representatives were considered in more detail – Moodle and WebCT. The analysis showed that these two products have their advantages and disadvantages, which helped a lot in the design of the system.

What is the system's life cycle?

After the analysis of three systems development life cycles, it was decided to use a spiral model because it gives more freedom to the developer and provides flexibility in interaction with potential customers.

What are the system requirements?

The following system requirements were formulated for the application under development:

- System Requirements
- Structure and System Operation Requirements
- Compatibility Requirements
- Operation Modes Requirements
- Reliability Requirements
- Ergonomic and Technical-Aesthetic Requirements
- Data Protection Requirements

What are the data structure requirements for the system?

The application uses a large array of data that is stored in a relational database. The database contains nine tables: users, news, groups, disciplines, group_access, lectures, attachments, subscriptions, and group_progress.

Which sections are available in the system?

The following sections are available in the system:

- Registration and Authorization section – provides authentication and the ability to create a new account.
- News section – the main page of the site, which shows the latest news.
- My courses section – contains a list of courses available to the user.
- My groups section – only available to teachers and administrators, and it is intended for the management of groups.
- Access rights section – available only to system administrators, and is used to create a new user with the necessary level of rights and manage roles of existing administrators and teachers in the system.

What is the process of developing a single-page application for an online learning environment?

It is the main research question of this thesis and in order to answer it, the process of single-page application development was shown. The following stages are highlighted:

- Stage 1. Planning

  At this stage, the application architecture is explained. Stage 1 is also covering all the corresponding technologies used in the system.

- Stage 2: Risk analysis and requirements

  At the Stage 2, there is an analysis of existing solutions on the market. Also, this stage reviews the system requirements and data structure requirements.

- Stage 3: Design and prototyping

  Stage 3 discusses the implementation of the required functionality as well as the appearance of the application and available sections. At this stage, there is a description of the working prototype and demonstration of the basic idea of how the application looks and works.

5.2 Suggestions and Recommendations for Further System Improvement

At the moment, the system has only basic functionality. Teachers can create courses, set deadlines, and manage groups. Students in turn can see all the assignments, but at this point they cannot upload their homework to the system and there are no tests for them to check their knowledge. Therefore, it is possible to add uploading of homework and tests in the future.

LIST OF REFERENCES

Printed Sources

Beighley, L. & Morrison, M. 2009. Head First PHP & MySQL. Sebastopol, CA: O'Reilly Media, Inc.

Freeman, E., Robson, E., Bates, B., & Sierra, K. 2004. Head First Design Patterns. Sebastopol, CA: O'Reilly Media, Inc.

Gamma, E., Vlissides, J., Helm, R., & Johnson, R. 1994. Design Patterns: Elements of Reusable Object-Oriented Software. Boston, MA: Addison-Wesley.

MacDonald, M. 2013. HTML5: The Missing Manual. 2nd Edition. Sebastopol, CA: O’Reilly Media, Inc.

Meyer, E. 2000. Cascading Style Sheets: The Definitive Guide. 1st edition. Sebastopol, CA: O'Reilly Media, Inc.

Zandstra, M. 2016. PHP Objects, Patterns, and Practice. 5th edition. Liverpool, UK: Apress.

Electronic Sources

About Moodle FAQ. 2019. About Moodle FAQ. Moodle [accessed 23 October 2020]. Available at: https://docs.moodle.org/39/en/About_Moodle_FAQ

Angelo, J. 2002. Courseware costs soar: blackboard and WebCT rethink pricing models. University Business [accessed 27 October 2020]. Available at: https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENPRESS/U020500A.pdf

AngularJS Tutorial. 2020. AngularJS Tutorial. TutorialsTeacher [accessed 19 October 2020]. Available at: https://www.tutorialsteacher.com/angularjs/angularjs-tutorials

Boehm, B. 2000. Spiral Development: Experience, Principles, and Refinements. Software Engineering Institute [accessed 26 October 2020]. Available at: https://resources.sei.cmu.edu/asset_files/SpecialReport/2000_003_001_13655.pdf

Bootstrap 3 Tutorial. 2020. Bootstrap 3 Tutorial 2020. W3Schools [accessed 19 October 2020]. Available at: https://www.w3schools.com/bootstrap/default.asp

Codesido, I. 2009. What is front-end development? The Guardian [accessed 17 October 2020]. Available at: https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost

CSS Tutorial. 2020. CSS Tutorial. W3Schools [accessed 18 October 2020]. Available at: https://www.w3schools.com/css/default.asp

Datatilsynet. 2017. Software development with Data Protection by Design and by Default. Datatilsynet [accessed 3 November 2020]. Available at: https://www.datatilsynet.no/en/about-privacy/virksomhetenes-plikter/innebygd-personvern/data-protection-by-design-and-by-default/?print=true

ED. 2011. Use of Technology in Teaching and Learning. ED [accessed 10 October 2020]. Available at: https://www.ed.gov/oii-news/use-technology-teaching-and-learning

Firesmith, D. 2013. Using V Models for Testing. Carnegie Mellon University [accessed 25 October 2020]. Available at: https://insights.sei.cmu.edu/sei_blog/2013/11/using-v-models-for-testing.html

GeeksforGeeks. 2020. Frontend vs Backend. GeeksforGeeks [accessed 15 October 2020]. Available at: https://www.geeksforgeeks.org/frontend-vs-backend/

Gilbertson, S. 2012. It's Official, CSS Media Queries Are a Web Standard. WIRED [accessed 18 October 2020]. Available at: https://www.wired.com/2012/06/its-official-css-media-queries-are-a-web-standard/

Harwell, D. 2020. Thousands of Zoom video calls left exposed on open Web. The Washington Post [accessed 10 October 2020]. Available at: https://www.washingtonpost.com/technology/2020/04/03/thousands-zoom-video-calls-left-exposed-open-web/

HTML Tutorial. 2020. HTML Tutorial. W3Schools [accessed 18 October 2020]. Available at: https://www.w3schools.com/html/default.asp

Marick, B. 1999. New Models for Test Development. Exampler [accessed 25 October 2020]. Available at: http://www.exampler.com/testing-com/writings/new-models.pdf

Moodle architecture. 2018. Moodle architecture. Moodle [accessed 23 October 2020]. Available at: https://docs.moodle.org/dev/Moodle_architecture

MySQL 8.0 Reference Manual. 2020. MySQL 8.0 Reference Manual. Oracle Corporation [accessed 21 October 2020]. Available at: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

Oxagile. 2014. What is the Waterfall Model? Oxagile [accessed 25 October 2020]. Available at: https://www.oxagile.com/article/the-waterfall-model/

PHP Manual. 2020. PHP Manual. The PHP Group [accessed 20 October 2020]. Available at: https://www.php.net/manual/en/index.php

PHPixie Documentation. 2020. PHPixie Documentation. PHPixie [accessed 21 October 2020]. Available at: https://phpixie.com/docs.html

Plitnichenko, L. 2020. 10 Challenges of E-Learning during COVID-19. Jellyfish.tech [accessed 10 October 2020]. Available at: http://web.archive.org/web/20200918132644/https://jellyfish.tech/10-challenges-of-e-learning-during-covid-19/

Ponchon, C. 2015. Exclusive interview with Mr Xavier Prats Monné. EOSE [accessed 10 October 2020]. Available at: http://eose.org/2015/01/exclusive-interview-with-mr-xavier-prats-monne/

Reliability Basics. 2007. Reliability Basics. ReliaSoft Corporation [accessed 1 November 2020]. Available at: https://www.weibull.com/hotwire/issue80/relbasics80.htm

Royce, W. 1970. Managing the Development of Large Software Systems. The Institute of Electrical and Electronics Engineers [accessed 25 October 2020]. Available at: http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf

Sheldon, F. 1998. Requirements Engineering. University of Colorado Colorado Springs [accessed 30 October 2020]. Available at: https://www.csm.ornl.gov/~sheldon/cs531/ch5.pdf

Software Engineering Tutorial. 2020. Software Engineering Tutorial. Tutorials Point [accessed 28 October 2020]. Available at: https://www.tutorialspoint.com/software_engineering/index.htm

System Analysis and Design Tutorial. 2020. System Analysis and Design Tutorial. Tutorials Point [accessed 24 October 2020]. Available at: https://www.tutorialspoint.com/system_analysis_and_design/index.htm

W3C. 2017. HTML 5.2 is now a W3C Recommendation. W3C [accessed 18 October 2020]. Available at: https://www.w3.org/blog/news/archives/6696

WebCT Java Problems. 2008. WebCT Java Problems. London School of Economics and Science [accessed 27 October 2020]. Available at: https://web.archive.org/web/20070116051757/http://clt.lse.ac.uk/Help/VLE-Pilot/WebCT-6-Java-Install.php