Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Другий (магістерський)
(рівень вищої освіти)

Розробка компонентів інформаційної технології проєктування систем E-learning
(тема)

Виконала:
студентка *II* курсу, групи *ІТПм-22-1*

Лактіонова А. О.
(прізвище, ініціали)

Спеціальність *122 Комп'ютерні науки*
(код і повна назва спеціальності)

Тип програми *Освітньо-професійна*
(освітньо-професійна або освітньо-наукова)

Освітня програма *Інформаційні технології проектування*
(повна назва освітньої програми)

Керівник *проф. Безкоровайний В. В.*
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри системотехніки _____ *Гребеннік І. В.*
(підпис) (прізвище, ініціали)

2024 р.

Я як студентка ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

12.01.2024 *Лактіонова А. О.*

Кваліфікаційна робота оформлена у відповідності до вимог діючих стандартів та методичних вказівок.

Матеріали кваліфікаційної роботи не містять відомостей, що заборонені для опублікування у відкритих виданнях.

Попередній захист проведено.

Керівник кваліфікаційної роботи *В. В. Безкоровайний*

Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*

Кафедра *Системотехніки*

Рівень вищої освіти *другий (магістерський)*

Спеціальність *122 Комп'ютерні науки*

Освітня програма *Інформаційні технології проектування*

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«___» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентці *Лактіоновій Аліні Олександрівні*

1. Тема роботи «*Розробка компонентів інформаційної технології проєктування систем E-Learning»*
затверджена наказом по університету від *«20» листопада 2023 р. №13736 Ст.*
2. Термін подання студентом роботи до екзаменаційної комісії *«15» січня 2024 р.*
3. Вихідні дані до роботи *Об'єкт дослідження – процес дистанційного навчання.. Предмет дослідження – інформаційні технології проектування систем дистанційного навчання. Предмет розробки – компоненти інформаційної технології проєктування систем E-Learning. Мета кваліфікаційної роботи – підвищення ефективності систем дистанційного навчання шляхом розробки інформаційно-технологічних компонентів їх проектування. Технічне забезпечення: IBM-сумісний персональний комп'ютер.*
4. Перелік питань, що потрібно опрацювати в роботі *Вступ. Огляд та аналіз існуючих технологій проектування системи електронного навчання. Системи електронного навчання як об'єкти проєктування. Огляд існуючих технологій проектування систем. Ідентифікація проблеми і актуальність рішень. Багатокритеріальний вибір варіантів побудови системи. Уточнення мети та формулювання завдань кваліфікаційної роботи. Формулювання вимог до програмної системи. Постановка завдання щодо вдосконалення технології проектування системи. Бізнес-цілі та критерії успіху. Ділові ризики. Нефункціональні вимоги до програмного забезпечення електронного навчання. Обґрунтування архітектури та проектування програмного забезпечення. Проектування баз даних. Опис обраної інформаційної технології для проектування програмної системи. Рішення для технології проектування бази даних. Опис розроблених компонентів. Висновки. Перелік джерел посилання.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій *кресленики, схеми, плакати та/або комп'ютерні ілюстрації (слайди) на аркушах формату А4, що включаються до тексту пояснювальної записки або складу додатків: екранні форми існуючих систем дистанційного навчання; схеми та алгоритми роботи систем дистанційного навчання; схема технології проєктування; ER діаграма системи дистанційного навчання; схема нормалізації бази даних; екранні форми програми розробленого засобу.*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Отримання завдання на кваліфікаційну роботу*	*20.11.2023*	*Виконано*
2	*Огляд сучасних інформаційних технологій*		
	проєктування систем дистанційного навчання	*27.11.2023*	*Виконано*
3	*Постановка задачі на розробку системи*	*30.11.2023*	*Виконано*
4	*Вибір середовища розробки*	*27.11.2023*	*Виконано*
5	*Проектування та розробка системи*	*30.11.2023*	*Виконано*
6	*Підготовка публікацій за результатами*		
	дослідження	*01.12.2023*	*Виконано*
7	*Оформлення пояснювальної записки*	*25.12.2023*	*Виконано*
8	*Подання закінченої роботи науковому*		
	керівникові	*28.12.2023*	*Виконано*
9	*Усунення зауважень наукового керівника*	*05.01.2024*	*Виконано*
10	*Підготовка презентації*	*08.01.2024*	*Виконано*
11	*Подання роботи на рецензування*	*10.01.2024*	*Виконано*
12	*Попередній захист*	*12.01.2024*	*Виконано*
13	*Подання роботи до екзаменаційної комісії*	*15.01.2024*	*Виконано*

Дата видачі завдання *«20» листопада 2023* р.

Студент _____ *Лактіонова А. О.*
(підпис) (прізвище, ініціали)

Керівник роботи _____ *проф. Безкоровайний В. В.*
(підпис) (посада, прізвище, ініціали)

ABSTRACT

Master’s Thesis: 74 p., 15 fig., 4 app., 36 sources.

DATABASE, DISTANCE LEARNING, DESIGN INFORMATION TECHNOLOGY, E-LEARNING SYSTEM, SOFTWARE, API, SQL, NOSQL, ACID.

The object of research is the distance learning process.

The subject of research is information technology for designing distance learning systems.

The purpose of the qualification work is to increase the effectiveness of distance learning systems through the development of information technology components of their design.

Research methods: systematic approach, methods of structural analysis and modeling, methods of modern information technologies.

The results of the work make it possible to choose and improve the design technology of E-Learning environments, which will allow creating high-performance and reliable software systems that meet functional requirements and can be competitive on the market.

The emphasis on scalability, data integrity, performance, security, and reliability directly corresponds to the system's ability to provide a seamless and efficient experience for both teachers and students in creating, managing, and interacting with educational content.

The developed systems for automating the process of distance learning can be used in the educational process of secondary schools, institutions of higher education, and institutions that carry out professional training.

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 74 с., 15 рис., 4 дод., 36 джерел.

БАЗА ДАНИХ, ДИСТАНЦІЙНЕ НАВЧАННЯ, ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ, СИСТЕМА ЕЛЕКТРОННОГО НАВЧАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, API, SQL, NOSQL, ACID.

Об’єкт дослідження – процес дистанційного навчання.

Предмет дослідження – інформаційна технологія проєктування систем дистанційного навчання.

Мета кваліфікаційної роботи – підвищення ефективності систем дистанційного навчання за рахунок розробки компонентів інформаційної технології їхнього проєктування.

Методи дослідження: системний підхід, методи структурного аналізу й моделювання, методи сучасних інформаційних технологій.

Результати роботи дозволяють здійснювати вибір та удосконалення технології проектування середовищ E-Learning, що дозволить створювати високопродуктивні та надійні програмні системи, які задовольняють функціональним вимогам і можуть бути конкурентоспроможними на ринку.

Акцент на масштабованості, цілісності даних, продуктивності, безпеці та надійності безпосередньо відповідає здатності системи забезпечувати цілісну та ефективну роботу як для викладачів, так і для учнів у створенні, управлінні та взаємодії з навчальним контентом.

Розроблювані системи для автоматизації процесу дистанційного навчання можуть використовуватися у навчальному процесі середніх шкіл, закладів вищої освіти та закладах і установах, які здійснюють підвищення кваліфікації фахівців.

THE CONTENT

INTRODUCTION

Recent trends, particularly the global quarantine prompted by the coronavirus epidemic, have underscored the increasing importance of distance learning in our world. Distance learning systems, firmly established in education, leverage the accelerated internet speeds, providing opportunities for multimedia learning. Notably, social media platforms have also significantly impacted education, shaping and evolving it continuously.

Traditional forms of education, often costly, time-consuming, and with no guaranteed outcomes, stand in stark contrast to E-Learning – versatile solutions addressing myriad challenges. The fundamental desires for time and cost savings have become paramount, as physical presence is no longer a prerequisite for work; everything is accessible in the portable virtual world.

Every day, hundreds of thousands of children and students join online lessons, acquiring material that would typically be covered in traditional classes. Various online learning services are actively utilized worldwide, facilitating comfortable and rapid knowledge acquisition.

Furthermore, as online courses proliferate rapidly, more adults seeking professional development turn to E-Learning. Major academic institutions worldwide, including Harvard, the Massachusetts Institute of Technology (MIT), and Stanford, now offer students the opportunity to receive education by enrolling in online courses and earning necessary credits—courses available for free or with additional charges.

This trend is especially relevant for students in underdeveloped countries and third-world nations, offering them the chance to utilize education for improving prospects, securing good jobs, and leading better lives.

However, the accessibility of qualified courses and lectures often comes at a significant cost, hindering individuals from less affluent families. Nonetheless,

my software system is free and user-friendly, ensuring accessibility for distance learning [1-2]. The simple and intuitive interface is understandable even for individuals with limited familiarity with technology and distance learning methods.

Moreover, the success of E-Learning systems hinges on the underlying information technologies employed for their design. Well-designed E-Learning systems are flexible and capable of accommodating future enhancements, technological advancements, and changes in educational methodologies. This forward-looking approach ensures the longevity and relevance of the system in the ever-evolving landscape of education technology. The backbone of this project lies in technologies such as Java, TypeScript, Spring framework, React, Redux, and PostgreSQL. These technologies collectively enable the creation of a seamless, efficient, and user-friendly platform for both learners and educators.

Users will have the ability to browse available courses, enroll in educational programs, take tests and quizzes after completing a course, and track their progress. For educators, the platform allows uploading lectures, grouping them into courses, creating educational programs, providing access to specific users or entire groups, and conducting convenient assessments through tests after course completion. The integration of these cutting-edge technologies ensures a robust and dynamic E-Learning experience for all stakeholders involved.

The object of research is the distance learning process.

The subject of research is information technology for designing distance learning systems.

The purpose of the qualification work is to increase the effectiveness of distance learning systems through the development of information technology components of their design.

To achieve the goal, it is necessary:

– carry out a review and analysis of existing electronic learning systems and their design technologies;

– determine the purpose and tasks of the research;

– to formulate the requirements for the software system and the task of improving its design technology;

– define the architecture and design the software of the software system;

– propose solutions for database design technology.

Research methods: systematic approach, methods of structural analysis and modeling, methods of modern information technologies.

The results of the work make it possible to choose and improve the design technology of E-Learning environments, which will allow creating high-performance and reliable software systems that meet functional requirements and can be competitive on the market.

The emphasis on scalability, data integrity, performance, security, and reliability directly corresponds to the system's ability to provide a seamless and efficient experience for both teachers and students in creating, managing, and interacting with educational content.

The developed systems for automating the process of distance learning can be used in the educational process of secondary schools, institutions of higher education, and institutions that carry out professional training.

1 OVERVIEW AND ANALYSIS OF EXISTING E-LEARNING SYSTEM DESIGN TECHNOLOGIES

1.1 Overview and analysis of existing E-Learning systems

In the rapidly evolving landscape of contemporary life, where distance learning has become an integral part, the educational market is no longer confined to traditional classrooms. This paradigm shift extends its reach not only to students but also to individuals seeking to swiftly acquire new technology skills with minimal expenditure. This shift has given rise to a proliferation of competitors in the educational technology sector, each striving to introduce novel and original features to capture the attention of a wider user base.

Among these contenders, Udemy emerges as a significant player in this domain, as depicted in Figure 1.1 [3]. Udemy's prominence can be attributed to its diverse course offerings, providing a broad spectrum of knowledge and skills to learners across the globe. However, amidst its strengths, there exists a notable drawback – the high cost associated with highly specialized courses, which are made available exclusively to users with a Business account.

This particular drawback introduces a significant consideration into the landscape of online learning. While Udemy's expansive course catalog and accessibility are commendable, the exclusive pricing structure for specialized content could potentially limit the inclusivity of its offerings. As the demand for specialized skills continues to grow, addressing the affordability and accessibility aspects of such courses becomes pivotal for ensuring that the benefits of distance learning are accessible to a diverse and expansive audience.

In the context of this competitive market, the strategies and decisions made by online learning platforms, including Udemy, bear substantial implications for the broader narrative of educational accessibility, affordability, and inclusivity.

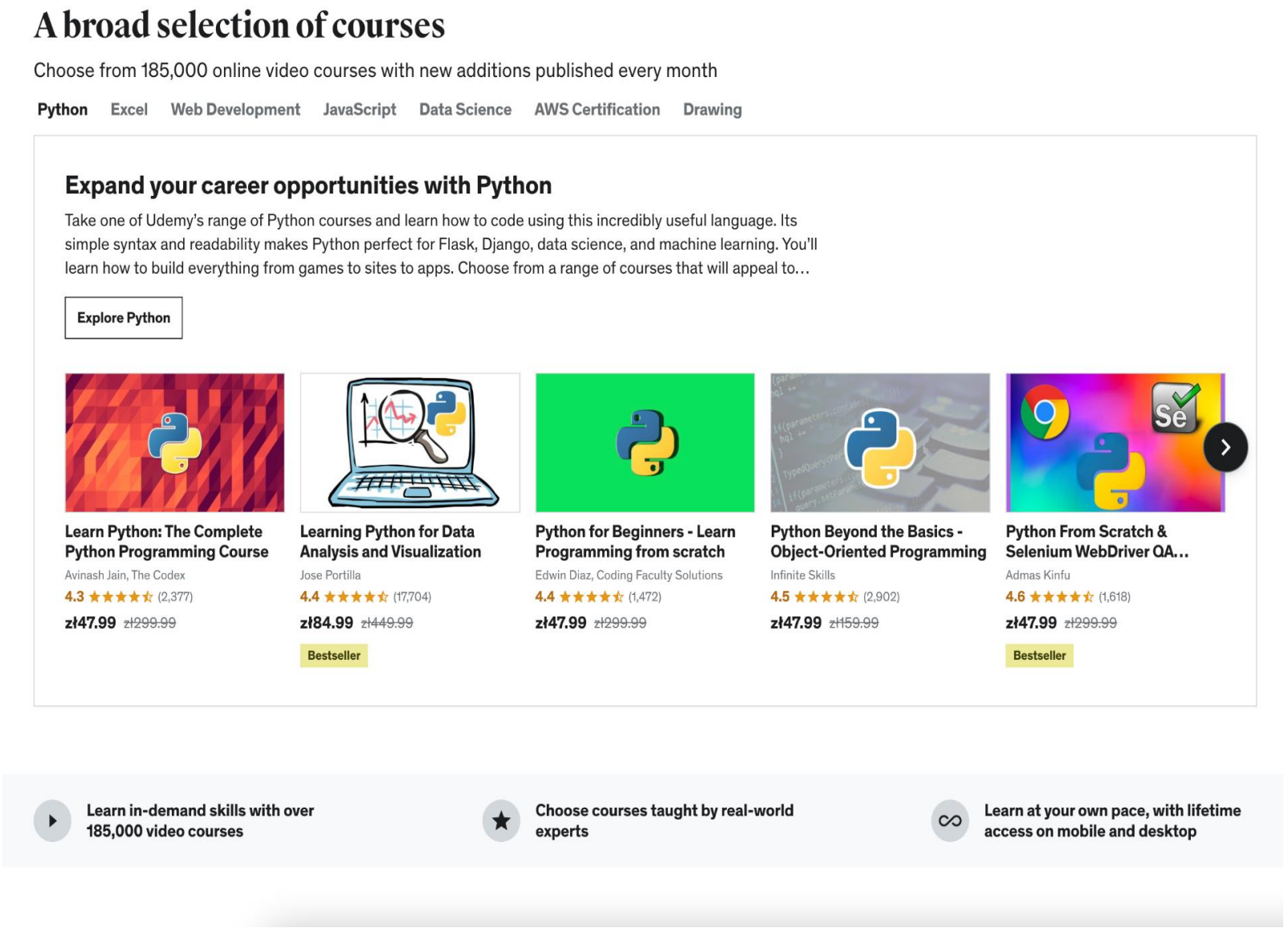

Figure 1.1 – Web-service «Udemy» [3]

The dynamics of this space continually evolve, and understanding the nuances of these shifts is imperative for both providers and learners alike. Each service strives to introduce novel and original features to attract a wider user base. The exploration and analysis of such considerations provide a valuable lens through which we can comprehend the current state and potential future trajectories of distance learning platforms in the ever-changing landscape of education.

Udemy is a platform empowering instructors to create online courses on diverse topics. Utilizing Udemy's course development tools, instructors can upload videos, PowerPoint presentations, PDF files, audio recordings, ZIP files, and conduct live sessions to craft comprehensive courses. Instructors can engage with users through online discussion boards as well. Courses span various categories, including business and entrepreneurship, science, arts, health and fitness, language, music, and technology. The majority of sessions focus on practical subjects, such as Excel software or utilizing iPhone cameras.

As of May 2022, the platform hosts over 52 million students, offering 196,000 courses taught by 68,000 instructors proficient in teaching in more than 75 languages. Students primarily enroll in courses to enhance job-related skills, with some courses providing credit towards technical certification.

Udemy has actively engaged corporate trainers seeking to develop coursework for their company's employees, reflecting its commitment to serving a diverse audience.

Udemy also offers Udemy for Business, providing enterprises with access to a curated collection of over 7 000 educational courses covering topics ranging from digital marketing tactics to office productivity, design, management, programming, and more.

Through Udemy for Business, organizations can create their own educational portals for corporate training (Figure 1.2).

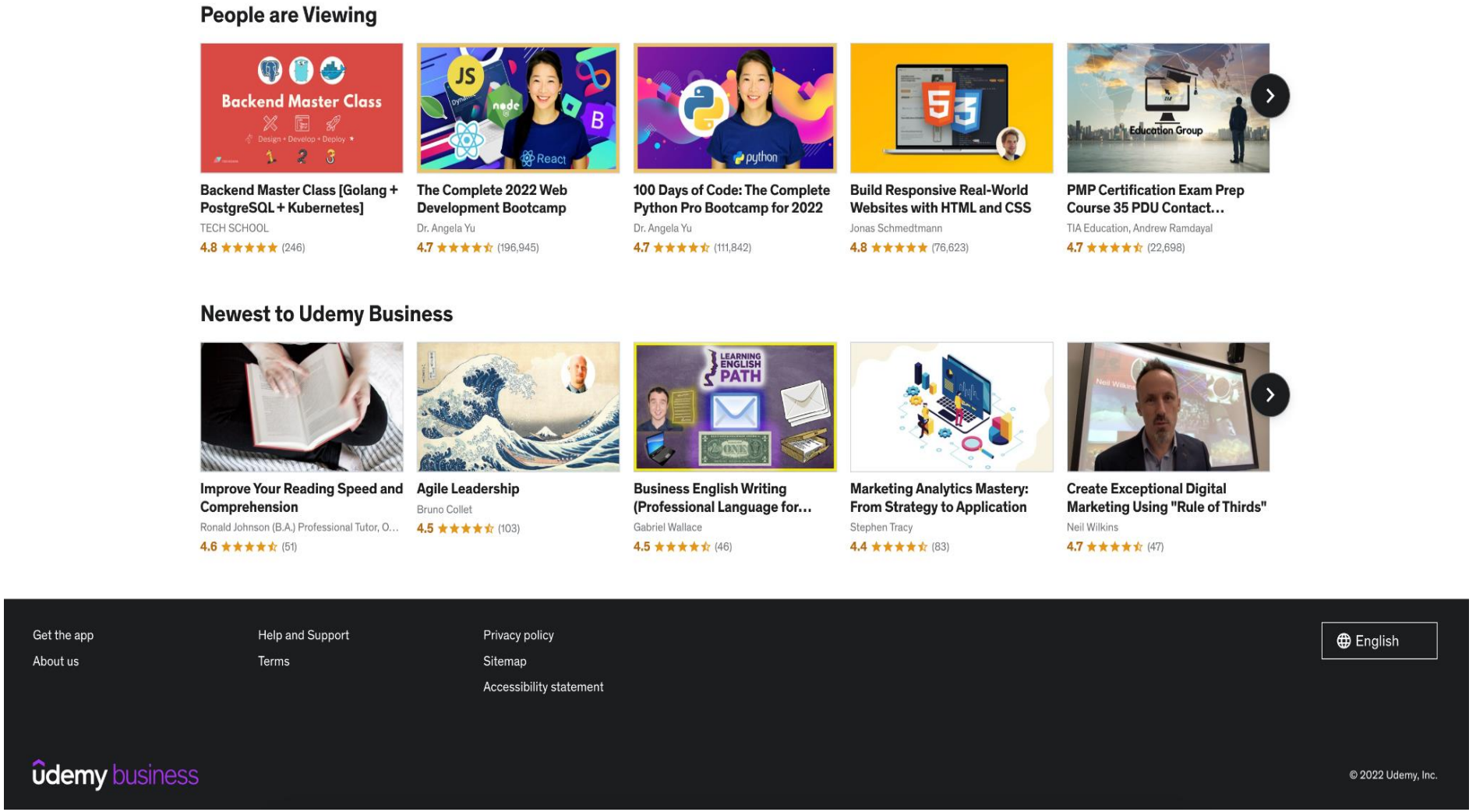

Figure 1.2 – Web-service «Udemy for business» [3]

An advantageous feature of the Udemy system is its mobile application (see Figure 1.3), downloadable on smartphones.

This allows users to study course materials more frequently and receive progress notifications and reminders. Another significant competitor is Google Classroom (Ukr. Google Клас) [4], a free web service developed by Google for educational institutions, aiming to simplify the creation, distribution, and classification of assignments in a paperless manner.

The primary goal of the service is to expedite the sharing of files between educators and learners. It is used by teachers and students in schools or higher education institutions by instructors and students. One of its main drawbacks is its reliance on Google services: Drive, Gmail.

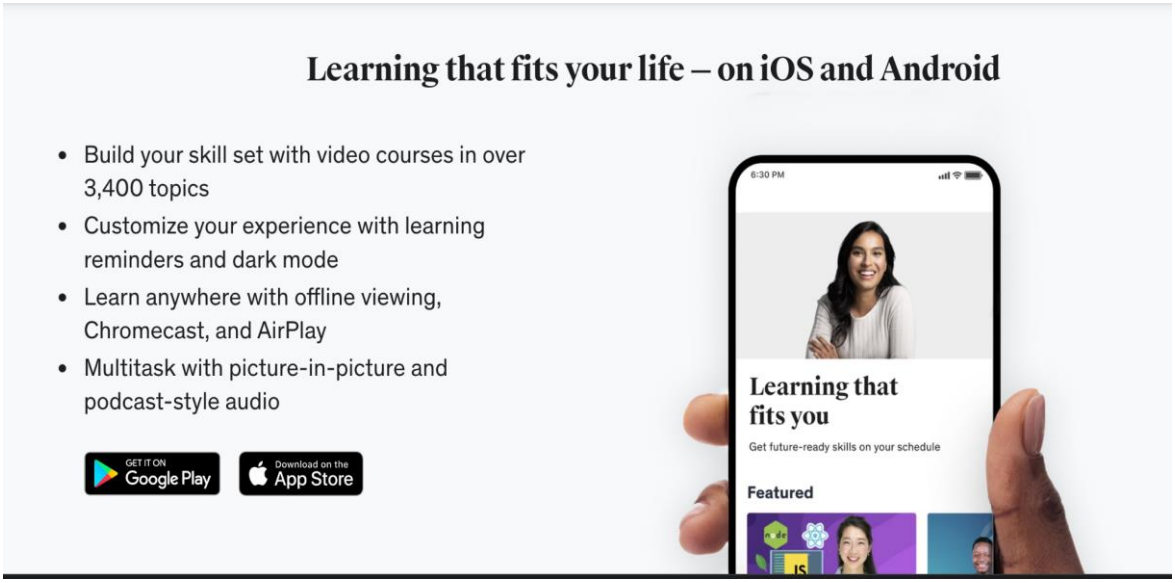

Figure 1.3 – Mobile app «Udemy» [3]

Google Classroom – one more pretty famous E-learning platform [5]. Google Classroom integrates various Google services, providing a comprehensive platform for educational institutions. It incorporates Google Drive for assignment creation and sharing, Google Docs, Sheets, and Slides for document creation, Gmail for communication, and Google Calendar for

scheduling, among other web services. Learners can be invited to classes via a private code or automatically imported from the school site. Each class generates a distinct folder on the user's Google Drive, where students submit work for assessment by the instructor. Mobile applications, available on iOS and Android, enable users to take photos, attach them to assignments, share files from other apps, and access information offline. Instructors can track each learner's progress, evaluate their work, and provide feedback.

Google Classroom, introduced in 2014, aims to streamline the educational process. It connects Google Drive, Docs, Sheets, Slides, Forms, Sites, Gmail, and other services, fostering a paperless environment (Figure 1.4). The integration of Google Calendar helps set assignment deadlines, plan trips, and schedule extracurricular activities.

The service allows students to join classes either through the educational institution's database, a private code, or automatic importation from the school site. Each Google Classroom-created class establishes a dedicated folder on the instructor's Google Drive, facilitating the submission and evaluation of assignments. For effective collaboration, assignments are stored and assessed within the Google suite, enabling interaction between teachers and students or among students themselves. Instead of exchanging documents directly, files are placed on the student's Google Drive, then submitted for assessment. Instructors can choose a file as a template, allowing each student to edit their own copy, subsequently returning it for evaluation.

Teachers have the option to post announcements in the class stream, encouraging student comments and fostering two-way communication. Students can also share materials in the class stream, although with lower priority than instructor announcements and subject to editing. Various types of Google product files, such as YouTube videos or Google Drive files, can be attached to announcements for information dissemination.

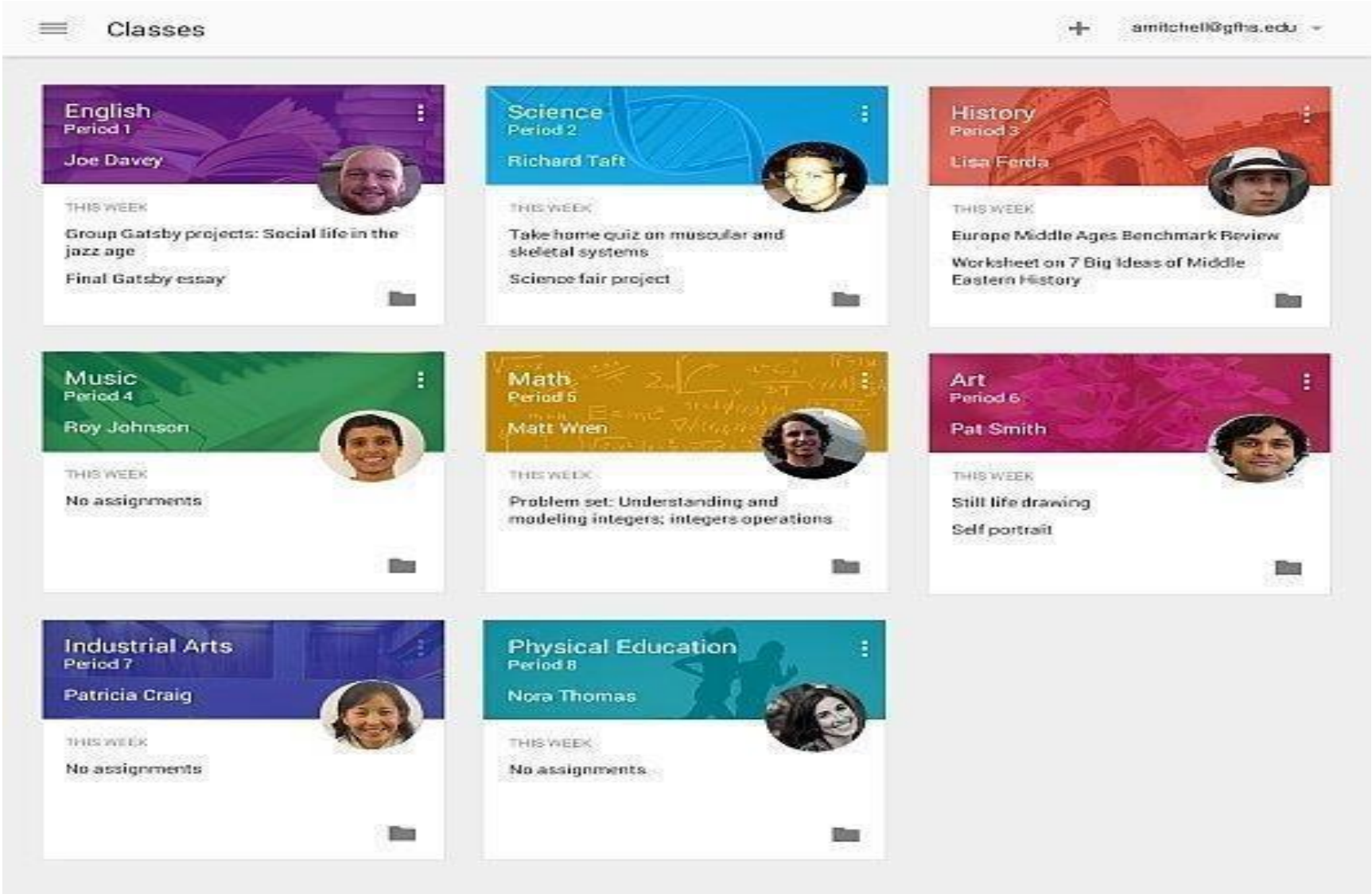

Figure 1.4 –Web-platform «Google Classroom» [4]

Gmail provides instructors with the ability to send emails to one or multiple students within the Google Classroom interface. Access to classes can be achieved through a web browser or the mobile Google Classroom applications on Android or iOS. Google Classroom allows teachers to archive courses at the end of a semester or year, removing them from the home page and placing them in the archive area to help organize classes. When a course is archived, both instructors and students can view it but cannot make any changes until it is restored.

The mobile application of Google Classroom, released in January 2015, is available for iOS and Android devices, enabling users to take photos, attach them to assignments, share files from other apps, and have offline access to them. Unlike Google's consumer services, Google Classroom, as part of G Suite for Education, does not display any ads in its interface, and user data is not scanned or used for advertising purposes.

In addition to already mentioned systems there is also Coursera, established in 2012, stands as a pioneering force in the realm of online education [6]. This platform, co-founded by Stanford University professors Andrew Ng and Daphne Koller, collaborates with esteemed universities and organizations globally to deliver a diverse array of courses, specializations, and degrees across various subjects (Figure 1.5).

Offering flexibility in learning, Coursera allows participants to access course materials at their own pace. The courses cover a wide spectrum of disciplines, including business, computer science, data science, health, and social sciences. Renowned professors and experts from top institutions create and teach these courses, ensuring high-quality educational content.

In addition to individual courses, Coursera features Specializations, which are a series of related courses designed to provide in-depth knowledge in specific fields. Completing a Specialization often involves a capstone project. Coursera's commitment to accessibility is reflected in its online degree programs offered in

collaboration with universities and colleges. These fully online degree programs provide structured curricula, culminating in a degree upon successful completion.

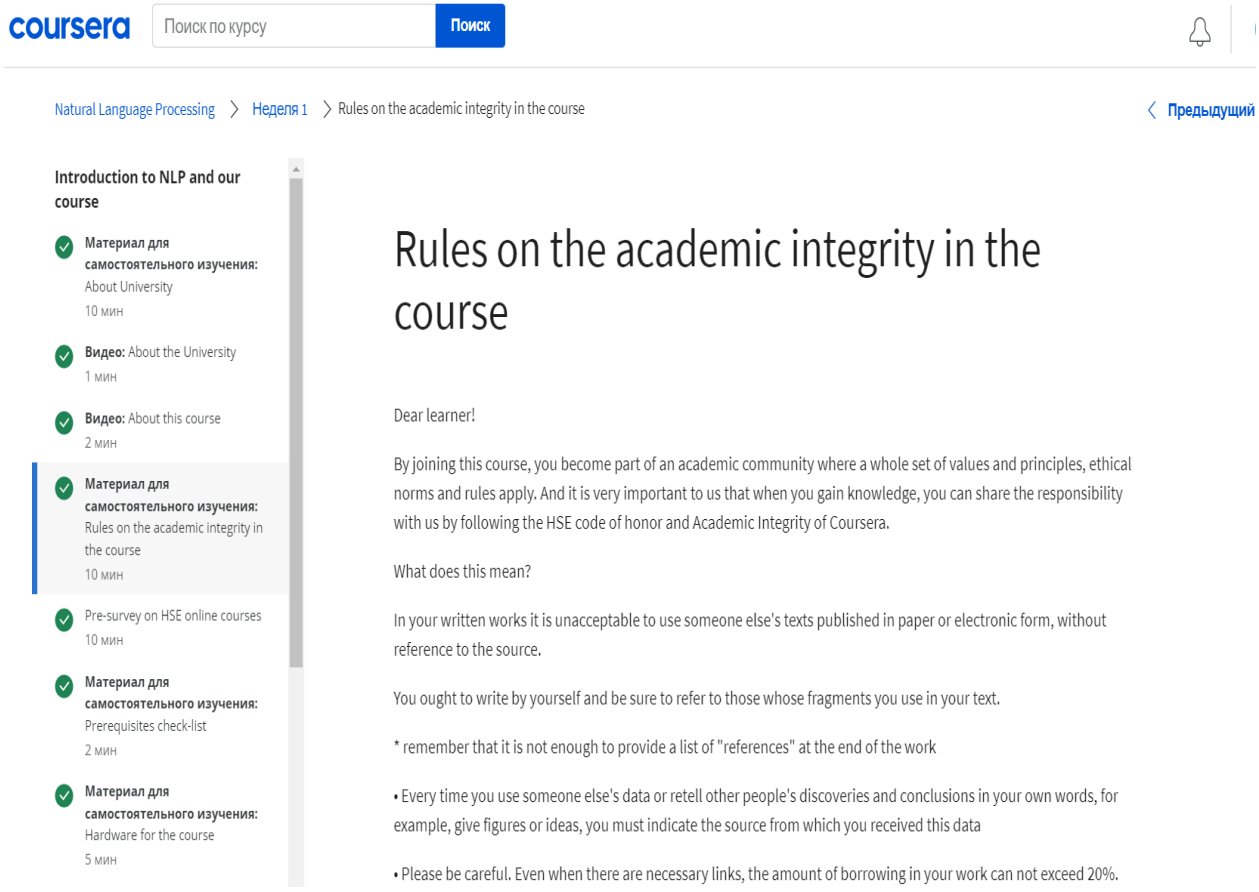

Figure 1.5 – Web-platform «Coursera» [6]

The platform facilitates interactive learning experiences through discussion forums, peer-reviewed assignments, and collaborative projects, fostering a sense of community among learners. Assessments, such as quizzes and assignments, are integral components of Coursera courses, with successful completion leading to certification.

Coursera's mobile accessibility, available through web browsers and dedicated iOS and Android applications, enables learners to access content conveniently from various devices.

Coursera Plus, the platform's subscription model, provides learners with

access to a vast library of courses and Specializations without additional costs for individual courses. Additionally, Coursera offers financial aid options to make high-quality education accessible to a diverse audience facing economic barriers.

In summary, Coursera has significantly contributed to the democratization of education, making top-notch learning opportunities available to a global audience through its innovative and inclusive online platform.

Talking about all E-learning systems and software systems in general, effective software design plays a pivotal role in addressing current challenges, such as scalability, user engagement, and content management. It provides the foundation for incorporating innovative features, improving accessibility, and adapting to evolving educational needs.

Moreover, a thoughtfully designed E-Learning system contributes to the success of online education initiatives by fostering a positive learning environment. It enables educators to create engaging content, monitor student progress, and tailor instructional strategies. Simultaneously, learners benefit from intuitive interfaces, personalized learning paths, and interactive features that enhance their overall educational experience.

1.2 Overview of existing system design technologies

System design is the process of designing the architecture and components of a software system to meet specific business requirements [7–9]. The process involves defining the system's architecture, components, modules, and interfaces, and identifying the technologies and tools that will be used to implement the system.

In order to successfully build a scalable system, we need to ensure that our

system is reliable, available, scalable, and maintainable (Figure 1.6) [10].

A system is Reliable when it can meet the end-user requirement. When you are designing a system you should have planned to implement a set of features and services in your system. If your system can serve all those features without wearing out then your System can be considered to be Reliable. A Fault Tolerant system can be one that can continue to be functioning reliably even in the presence of faults. Faults are the errors that arise in a particular component of the system. An occurrence of fault doesn't guarantee Failure of the System. Failure is the state when the system is not able to perform as expected. It is no longer able to provide certain services to the end-users.

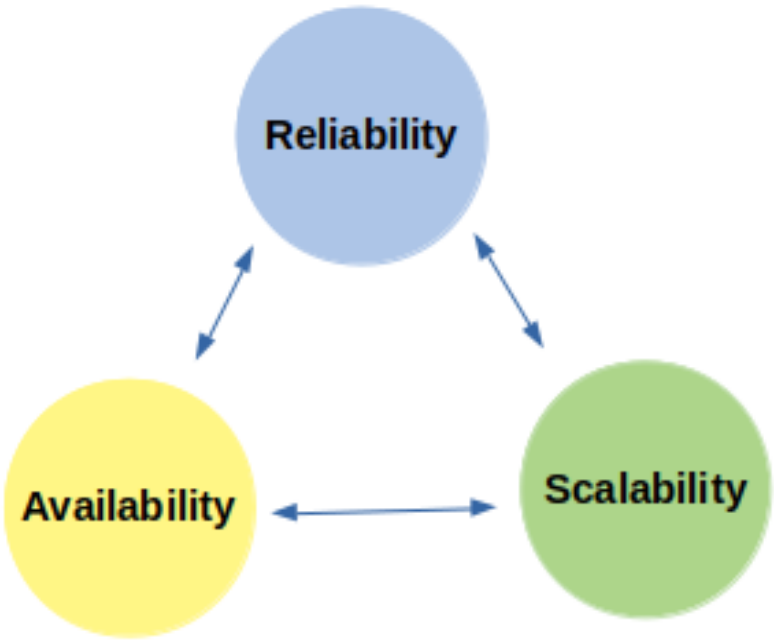

Figure 1.6 – Scalable system representation [10]

Availability is a characteristic of a System that aims to ensure an agreed level of Operational Performance, also known as uptime. It is essential for a system to ensure high availability in order to serve the user's requests. The extent

of Availability varies from system to system. Suppose you are designing a Social Media Application then high availability is not much of a need. A delay of a few seconds can be tolerated. Getting to view the post of your favorite celebrity on Instagram with a delay of 5 to 10 seconds will not be much of an issue. But if you are designing a system for hospitals, Data Centers, or Banking, then you should ensure that your system is highly available. Because a delay in the service can lead to a huge loss.

Scalability refers to the ability of the System to cope with the increasing load. While designing the system you should keep in mind the load experienced by it. It's said that if you have to design a system for load X then you should plan to design it for 10X and Test it for 100X. There can be a situation where your system can experience an increasing load. Suppose you are designing an E-commerce application then you can expect a spike in the load during a Flash Sale or when a new Product is Launched for sale. In that case, your system should be smart enough to handle the increasing load efficiently and that makes it Scalable. In order to ensure scalability you should be able to compute the load that your system will experience.

Navigating the complex terrain of information system software design reveals a diverse landscape shaped by a myriad of technological paradigms [11–12]. One such paradigm gaining prominence is the microservices architecture, a modern approach that involves breaking down large systems into smaller, autonomous services. This fosters flexibility, ease of development, and scalability, enabling organizations to adapt swiftly to evolving requirements.

Containerization, exemplified by Docker, stands as a linchpin in contemporary software design. It facilitates the encapsulation of applications and dependencies into containers, ensuring consistency across various environments. Docker simplifies deployment, enhances scalability, and seamlessly orchestrates the operation of microservices, contributing to the efficiency and agility of information systems.

Serverless computing emerges as a transformative model, allowing developers to focus on coding without the burden of managing servers. Cloud providers like AWS Lambda and Azure Functions automate execution and scaling, promising efficient resource utilization and cost-effectiveness.

DevOps practices have become integral to the software development lifecycle, fostering collaboration between development and operations teams. The aim is to automate and streamline processes, ensuring continuous integration and deployment, thereby enhancing the speed and quality of software delivery.

The integration of machine learning into software design is reshaping information systems by enabling data analysis, pattern recognition, and predictive modeling. This infusion empowers systems to learn, adapt, and optimize performance, revolutionizing decision-making processes.

GraphQL, a query language for APIs, offers an efficient means of communication between clients and servers. In contrast to traditional REST APIs, GraphQL allows clients to request only the necessary data, enhancing performance and responsiveness in information systems.

Blockchain technology is making inroads, particularly in sectors prioritizing security and transparency. Its decentralized and secure approach to recording and verifying transactions ensures data integrity and fosters trust in information systems handling sensitive data.

Low-code and no-code platforms are empowering users with varying technical expertise to participate in application development, accelerating the software creation process. These platforms, while increasing development speed, democratize software creation, allowing business users to contribute meaningfully.

Frameworks like React, Angular, and Vue.js dominate front-end development, facilitating the creation of interactive and responsive user interfaces. Single Page Application (SPA) architectures, enabled by these frameworks, contribute to faster load times and a more fluid user interface.

Cloud-native technologies are redefining information system design by leveraging cloud services, microservices architecture, and containerization. Platforms like Kubernetes orchestrate these cloud-native applications, providing a robust framework for managing containerized workloads.

In the fabric of my qualification work, these existing technologies represent not just tools but integral components in understanding and navigating the intricacies of software design for information systems. This exploration not only sheds light on the current state of the field but also sets the stage for innovative approaches and solutions in the ever-evolving landscape of information systems design.

1.3 Identification of issues and relevance of solutions

In the expansive realm of software development, the meticulous examination of challenges and the quest for relevant solutions in system design technologies constitute a vital aspect of advanced research. This comprehensive exploration is underscored by its pivotal role in determining the efficacy and sustainability of software applications within the dynamic technological landscape.

The linchpin in the development of viable software solutions is effective system design. The identification and resolution of challenges intrinsic to design technologies are paramount, as they directly influence the reliability, scalability, and adaptability of systems in real-world applications. Beyond the realm of mere functionality, well-designed systems play a significant role in enhancing the overall user experience, thereby impacting user satisfaction and engagement positively. Furthermore, in the face of rapid technological evolution, the imperative to address issues in system design technologies becomes even more pronounced, ensuring adaptability to emerging trends and seamless integration of

novel methodologies.

As we navigate through this exploration, the goal is not only to unravel the complexities but to offer tangible insights that can contribute to the continuous improvement and innovation in system design technologies. The synthesis of theoretical knowledge with practical implications serves as a foundation for developing solutions that not only address existing challenges but also anticipate and prepare for the future trajectory of software development. In essence, this examination goes beyond the immediate concerns, embracing a forward-thinking approach to foster resilience, agility, and excellence in software systems

Here is a list of Current challenges in technologies of system design for E-Learning platforms.

1. Scalability Challenges: E-Learning platforms often face hurdles in seamlessly accommodating a growing user base. Traditional system designs may struggle with performance bottlenecks, slower response times, and potential downtime during peak usage periods.

2. Integration Complexities: diverse components within E-Learning systems, such as content management systems and learning management systems, often pose integration challenges. Incompatibilities between technologies can lead to data silos and fragmented user experiences.

3. Security and Privacy Concerns: security remains a critical concern, with the sensitive nature of user data and the prevalence of online assessments. Data breaches, unauthorized access, and vulnerabilities in the system's architecture pose significant threats.

4. Lack of Personalization: many E-Learning systems struggle to provide personalized learning experiences tailored to individual students, impacting the diverse learning styles, paces, and preferences of users.

5. Limited Interactivity and Engagement: effective learning demands active engagement, but some E-Learning systems may fall short in providing interactive and engaging content, diminishing overall user experiences.

6. Technological Obsolescence: rapid advancements can render E-Learning systems obsolete if they are not designed with adaptability in mind, limiting their ability to support new content formats, emerging devices, or innovative pedagogical approaches.

7. Accessibility Challenges: ensuring accessibility for users with diverse needs, including those with disabilities, remains a challenge. Some E-Learning platforms may not adhere to universally accepted accessibility standards.

8. Content Management and Updates: managing and updating educational content efficiently is a recurrent challenge, impacting the relevance and reliability of learning material.

9. Vendor Lock-In Issues: E-Learning platforms may lead to vendor lock-in, making it challenging to migrate to alternative systems due to proprietary formats or dependencies.

10. Inadequate Analytics and Reporting: comprehensive analytics are crucial for tracking student progress, assessing course effectiveness, and making data-driven improvements. Some E-Learning systems may lack robust analytics features.

Addressing these challenges involves innovative approaches in system design, ushering in a new era for E-Learning platforms. I explored the compelling relevance of advanced solutions in overcoming the aforementioned issues which are listed below.

1. Future-Proofing Systems: effective solutions to identified issues ensure the future-proofing of software systems. By anticipating and addressing scalability, integration, and security challenges, design technologies contribute to the longevity and sustainability of software applications.

2. Catalyzing Innovation: robust design solutions empower developers to innovate. By overcoming current challenges, design technologies provide a foundational framework for integrating innovative features, technologies, and methodologies into software systems.

3. User-Centric Paradigm: solutions aimed at enhancing user-centric design contribute to surpassing user expectations. Addressing issues related to user interface, experience, and accessibility ensures that software applications resonate with end-users.

4. Agile-Friendly Foundations: design solutions that seamlessly integrate with agile methodologies contribute to the efficiency and adaptability of software development processes. This relevance ensures that software projects can navigate changes, iterations, and evolving requirements with agility.

5. Elevating Security Measures: given the escalating threats in cybersecurity, solutions addressing security concerns are imperative. Design technologies must incorporate robust security measures, safeguarding software applications and the sensitive data they handle.

In conclusion, the meticulous examination of challenges and the pursuit of relevant solutions in system design technologies are pivotal endeavors within the realm of advanced research. Through my work, I aspire to pioneer dynamic, inclusive, and impactful online learning experiences for students worldwide. This exploration not only informs the development of resilient and innovative software but also contributes significantly to the evolving narrative of user-centric, secure, and adaptable systems in contemporary software design.

1.4 Multi-criteria selection of system construction options

The decomposition scheme of the system composition optimization problem is presented in the form of a set of subproblems and individual tasks $Task$ of different levels with connections by input data and solution results [13]:

$$Problem = \{ SubProblem^l \},\ SubProblem^l = \{ Task_i^l \},\ l = \overline{1,n_l},\ i = \overline{1,i_l}, \qquad (2.1)$$

where n_l is the number of levels of decomposition;

$Task_i^l$ – the i-th task of the l-th level;

i_l is the number of tasks at the l-th level.

The first level of decomposition includes a complex of the following sub-problems: $SubProblem_1$ – formation of requirements for the system and development of the technical task for its creation; $SubProblem_2$ – system development; $SubProblem_3$ – adaptation of the system; $SubProblem_4$ – system reengineering.

At the lower level of decomposition, a set of elementary works is distinguished, which are considered as separate local tasks of the development of components $\{Task_i\}$, $i = \overline{1,n}$. Such tasks are solved cyclically, taking into account the requirements for the system and its quality indicators:

– necessary functionality of the system;

– requirements for the used technical means (desired frequency of the processor, required memory volumes, etc.);

– the level of necessary knowledge of the user;

- the cost of the system, etc.

Similar requirements can be considered as limitations or local criteria of system efficiency $k_j(s)$, $j = \overline{1,m}$ (where s is the system construction option; m is the number of local criteria for evaluating system properties).

Existing technologies for supporting project decision-making are based on the ranking of acceptable system construction options $S = \{s\}$ using cardinal and ordinal approaches [13–18].

Within the framework of the cardinalistic approach, preferences between system quality indicators can be specified by the values of the weighting coefficients λ_j of local criteria $k_j(s)$, $j = \overline{1,m}$. In the presence of local criteria

that have different directions $k_i(s) \to min$ and $k_j(s) \to max$ use their utility functions $\xi_j(s),\ j = \overline{1,m}$:

$$\xi_j(s) = k_j(s) - k_j^-)/(k_j^+ - k_j^-), \tag{1.2}$$

where $k_{j(s)},\ j = \overline{1,m}$ is the value of the j-th local criterion for the s-th variant of system construction;

k_j^+ – the best value of the j-th local criterion;

k_j^- is the worst value of the j-th local criterion.

The best option for building the system will correspond to the maximum of the additive function of total utility:

$$s^o = arg \max_{s \in S} \left\{ P(s) = \sum_{j=1}^{m} \lambda_j \xi_j(s) \right\}. \tag{1.3}$$

At the same time, the weighting coefficients of local criteria $\lambda_j \geq 0$, $j = \overline{1,m}$ must satisfy the normalization condition $\sum_{j=1}^{m} \lambda_j = 1$.

In a situation where the numerical values of the weighting factors cannot be determined quantitatively, it is proposed to order the local criteria according to the degree of their importance $k_1 \succ k_2 \succ ... \succ k_m$.

Then, to choose the best version of the system, it is suggested to use the method of lexicographic optimization. The option of building a system s is lexicographically better than the option v if one of the conditions is met:

$$\xi_1(s) > \xi_1(v);$$

$$\xi_1(s) = \xi_1(v),\ \xi_2(s) > \xi_2(v);$$

…

$$\xi_j(s)=\xi_j(v),\ j=\overline{m-1},\ \xi_m(s)>\xi_m(v).$$

At the first stage, it is necessary to find a subset of options $S_1^o \subseteq S$ that are optimal according to the local criterion $k_1(s)$.

At the second stage, it is necessary to find a subset of options $S_2^o \subseteq S_1^o$ that are optimal according to the local criterion $k_2(s)$, and so on.

At the last stage, the best system construction option $s^o \in S_{m-1}^o$ is selected from the subset of options $S_{m-1}^o \subseteq S_{m-2}^o$ that are optimal according to the criterion $k_m(s)$.

If when choosing according to local criteria $k_j(s),\ j=\overline{m-1}$, only one option will be selected, the corresponding subset $S_j^o,\ j=\overline{m-1}$ must be expanded at the expense of quasi-optimal options for building the system.

It should be taken into account that too small a concession size (deviation from the best value) does not allow taking into account the value of all local quality criteria of the system, and too large – to take into account the given ordering of criteria.

1.5 Definition of the purpose of the qualification work

In the ever-evolving landscape of contemporary education, the optimization of E-Learning software systems stands as a pivotal endeavor, forming the crux of this qualification work investigation. The imperative for this research stems from the transformative shift towards remote education, a global phenomenon further underscored by recent events. The focus is not merely on the integration of technology but on the nuanced refinement of systems to meet the

diverse needs of modern learners.

At the heart of this inquiry lies the recognition of E-Learning as a cornerstone of accessible and engaging education. As traditional boundaries dissolve in the wake of global challenges, understanding and enhancing the efficiency of these systems becomes paramount for fostering inclusivity. This research seeks to delve into how well-designed E-Learning systems can transcend geographical constraints, fostering global collaboration and ensuring an inclusive educational environment for learners from various backgrounds.

The qualification work aims to unfold the narrative of flexibility inherent in E-Learning. It goes beyond the technological aspects, exploring how efficient systems empower learners to engage with educational content on their own terms. This investigation is not a mere exploration of software design principles but an inquiry into how these principles can amplify the flexibility of E-Learning, accommodating diverse learning preferences and schedules.

In parallel, the research acknowledges the integral role of technological advancements in shaping the educational landscape. The qualification work endeavors to unravel the symbiotic relationship between sophisticated software design and emerging technologies such as artificial intelligence and augmented reality. It seeks to understand how these innovations can be seamlessly integrated, elevating the overall learning experience for users.

As learning preferences and trends continue to evolve, the qualification work recognizes the need for E-Learning systems to be adaptable. This research explores the agility of efficient systems, investigating how they can embrace new methodologies and technologies to meet the dynamic expectations of modern learners. The inquiry extends beyond theoretical considerations, aiming to offer practical insights into creating E-Learning environments that are both responsive and relevant.

The economic implications of optimizing E-Learning systems are integral to this research. It aims to explore how improved efficiency contributes to cost-

effectiveness, ensuring optimal resource utilization. This includes reducing costs associated with infrastructure, materials, and even travel, making education more accessible to a broader audience.

In the professional realm, continuous learning has become imperative. This qualification work investigates how efficient E-Learning systems play a pivotal role in supporting workforce development. It aims to understand how these systems provide a platform for lifelong learning and skill enhancement, aligning educational endeavors with the evolving demands of the professional landscape.

Furthermore, the research delves into the significance of data-driven decision-making in E-Learning systems. It emphasizes how insights derived from analytics can inform educational processes, refining and personalizing learning experiences. This approach is not just a theoretical consideration but a practical exploration of how data can be leveraged to enhance educational outcomes.

In times of global challenges, the resilience of E-Learning systems becomes evident. This qualification work aims to unravel how robust systems enable educational institutions to navigate disruptions seamlessly, ensuring uninterrupted and high-quality education. It seeks to position E-Learning not just as a response to crises but as a resilient and integral component of the educational ecosystem.

However, my focus is specifically directed towards the critical decision between SQL [19] and NoSQL [20] databases. In the foundational architecture of E-Learning systems, the selection between SQL and NoSQL databases holds a pivotal role in shaping their structure, performance, and adaptability. In the forthcoming analysis, I will delve into the examination of these two prominent database paradigms, exploring their implications for data management in E-Learning systems. The significance of this exploration lies in the central role databases play in storing, retrieving, and managing data, thereby influencing the overall efficiency, scalability, and adaptability of the system.

The decision-making process in selecting the appropriate database

technology carries substantial implications for the effectiveness and sustainability of E-Learning platforms. In the dynamic landscape of educational technology, the right choice can enhance data organization, retrieval speed, and contribute to the overall success of the E-Learning ecosystem.

As I embark on this analytical journey, the objective is not merely to scrutinize technologies but to discern their impact on the holistic design and functionality of E-Learning systems. The nuanced understanding gained from this exploration will serve as a guide for decision-makers, steering them towards well-informed choices that align with the evolving needs and expectations of modern educational environments. Through this focused investigation into database selection, I aim to contribute valuable insights that will resonate in the broader discourse of software architecture for E-Learning systems [21].

The object of research is the distance learning process.

The subject of research is information technology for designing distance learning systems.

The purpose of the qualification work is to increase the effectiveness of distance learning systems through the development of information technology components of their design.

To achieve the goal, it is necessary:

– to formulate the requirements for the software system and the task of improving its design technology;

– define the architecture and design the software of the software system;

– propose solutions for database design technology.

2 FORMULATION OF REQUIREMENTS FOR THE SOFTWARE SYSTEM

2.1 Task definition for improving the technology of the system design

In delineating the path towards enhancing the technology of system design, the following tasks are outlined for meticulous consideration and systematic execution.

1. Comprehensive analysis of existing technologies. Undertake an in-depth examination of the current landscape of system design technologies, identifying strengths, weaknesses, and emerging trends. This analysis will serve as the foundation for informed decision-making in the pursuit of improvement.

2. Benchmarking and comparative evaluation. Conduct benchmarking exercises to assess the performance, scalability, and efficiency of existing technologies. A comparative evaluation against industry standards will provide valuable insights into areas requiring enhancement and potential opportunities for innovation.

3. Identification of system design challenges: Delve into the intricacies of the current system design, pinpointing challenges and bottlenecks. This step is crucial in understanding the specific pain points that necessitate improvement, ensuring that the enhancements directly address real-world issues.

4. Stakeholder collaboration and feedback gathering: Engage with stakeholders, including end-users, developers, and administrators, to gather comprehensive feedback on the existing system. Collaborative sessions and feedback loops will provide firsthand insights into user experiences, expectations, and areas of dissatisfaction.

5. Define clear objectives for technological improvement. Formulate precise and measurable objectives for enhancing system design technology.

These objectives should align with organizational goals, user expectations, and the broader context of technological advancements in the industry.

6. Perform testing and optimization. Test how does design decision meet project needs and identify areas for optimization and fine-tune the technology stack to achieve optimal results, mitigating potential performance bottlenecks.

2.2 Business goals and success criteria

In the dynamic realm of E-Learning systems, the strategic selection of technologies for software design plays a pivotal role in shaping business outcomes and achieving overarching goals. The alignment of technology choices with business objectives is not merely a technical decision but a crucial determinant of success. Here, I delineate the business goals and success criteria that underpin the judicious selection of technologies for the software design of our E-Learning system.

Business Goals.

1. Scalability and Performance

Goal: Ensure the E-Learning system can seamlessly scale to accommodate growing user bases.

Rationale: Facilitate business growth by handling increased demand without compromising performance.

2. User Engagement and Experience

Goal: Enhance user engagement through an intuitive and immersive learning experience.

Rationale: Improve user satisfaction, retention, and overall effectiveness of the E-Learning platform.

3. Adaptability and Future-Readiness

Goal: Build a system that can easily adapt to emerging technologies and evolving educational methodologies.

Rationale: Future-proof the E-Learning system, ensuring relevance and longevity in a rapidly changing technological landscape.

4. Cost-Efficiency

Goal: Optimize costs associated with development, maintenance, and scalability.

Rationale: Ensure a sustainable business model by minimizing operational expenses while maximizing returns.

5. Security and Data Integrity

Goal: Safeguard user data and maintain the integrity of educational content.

Rationale: Uphold user trust, comply with data protection regulations, and mitigate risks associated with cyber threats.

Success Criteria:

1. Scalability Metrics

Criteria: Achieve a specified percentage increase in the number of concurrent users without performance degradation.

Measurement: Regular load testing and monitoring during peak usage periods.

2. User Satisfaction Index

Criteria: Attain a target score in user satisfaction surveys and feedback mechanisms.

Measurement: Periodic surveys, user feedback analytics, and qualitative assessments.

3. Technology Adoption Rate

Criteria: Ensure a smooth transition and high adoption rate of new features and technologies.

Measurement: Tracking the rate of adoption for new system features and technologies.

4. Cost Reduction Metrics

Criteria: Realize a defined reduction in overall development and maintenance costs.

Measurement: Regular financial audits and cost-benefit analyses.

5. Security and Compliance

Criteria: Maintain a zero-incident record of security breaches and ensure compliance with data protection regulations.

Measurement: Continuous monitoring, regular security audits, and compliance assessments.

By aligning technology choices with these business goals and establishing clear success criteria, the selection and implementation of software design technologies for the E-Learning system become strategic drivers of business success. This approach ensures that technology decisions are not made in isolation but are integral to achieving the broader objectives of the E-Learning platform.

2.3 Business risks

The interplay between technology and business objectives is delicate, and understanding the potential risks is crucial for informed decision-making. Below, I listed the key business risks associated with the technology of software design for our E-Learning system.

1. Scalability Challenges.

Risk: Inadequate scalability of chosen technologies may lead to system bottlenecks and poor performance as user demand increases.

Impact: Reduced user satisfaction, potential loss of users, and compromised business growth.

2. Compatibility Issues.

Risk: Incompatibility between chosen technologies and existing systems may hinder seamless integration, causing disruptions and inefficiencies.

Impact: Increased development costs, delayed timelines, and potential user dissatisfaction due to system inconsistencies.

3. Dependency on Third-Party Components.

Risk: Relying heavily on third-party components may expose the system to vulnerabilities, downtime, or changes in external APIs.

Impact: Potential security breaches, system instability, and compromised user trust.

4. Rapid Technological Obsolescence.

Risk: Swift advancements in technology may render chosen design technologies obsolete, necessitating frequent updates and adaptations.

Impact: Increased development and maintenance costs, potential disruptions, and challenges in keeping up with industry standards.

5. Security Vulnerabilities.

Risk: Inherent security flaws or vulnerabilities in chosen technologies may expose the E-Learning system to data breaches and privacy concerns.

Impact: Damage to the organization's reputation, legal repercussions, and loss of user trust.

6. Skill Gaps and Talent Availability.

Risk: Choosing niche or less commonly used technologies may result in a scarcity of skilled developers, impacting system development and maintenance.

Impact: Increased labor costs, delays in project timelines, and potential compromise in code quality.

7. Unforeseen Integration Challenges.

Risk: Complex integration requirements with other business systems may pose challenges that were not initially anticipated.

Impact: Delays in project timelines, increased development costs, and

potential disruptions to existing business processes.

8. Regulatory Compliance Concerns.

Risk: Failure to comply with evolving data protection and privacy regulations may result in legal consequences and financial penalties.

Impact: Legal liabilities, reputational damage, and financial losses due to non-compliance.

Addressing these risks requires a proactive risk management strategy, including continuous monitoring, thorough testing, and contingency planning. By acknowledging these potential challenges, the business can adopt a resilient stance, ensuring that the technology choices made for the software design of the E-Learning system contribute positively to business objectives while mitigating potential risks.

2.4 Non-functional requirements for designing E-Learning software system

Non-functional requirements are crucial aspects of system development that focus on characteristics beyond the core functionality. They define how well a system performs certain functions rather than what functions it performs. In the context of an E-Learning system, various non-functional requirements play a pivotal role in ensuring the system's reliability, performance, and overall user experience. Here's an exploration of key non-functional requirements for designing E-Learning systems.

1. Performance: the system should respond to user actions within a maximum of 2 seconds, ensuring a seamless and efficient user experience.

2. Availability: the system should be available 99.9% of the time during standard operating hours, providing continuous access for learners and

instructors.

3. Security: user authentication must be performed using secure protocols (e.g., HTTPS) with encryption, ensuring the protection of sensitive educational information.

4. Scalability: the system should be scalable to accommodate a 20% increase in concurrent users within a one-month period, ensuring optimal performance during periods of high user activity.

5. Usability: the user interface should follow accessibility standards, supporting screen readers and keyboard navigation for an inclusive learning environment.

6. Compatibility: the system should be compatible with the latest versions of commonly used browsers (Chrome, Firefox, Safari), ensuring a consistent user experience.

7. Reliability: the system should have a mean time between failures (MTBF) of at least 30 days, providing a stable learning environment with reduced system failures.

8. Compliance: the system should comply with data protection regulations (e.g., GDPR) and accessibility standards (e.g., WCAG), ensuring legal adherence and user privacy.

9. Interoperability: the system should support integration with external learning management systems through standardized APIs, fostering a cohesive and integrated learning environment.

10. Capacity: the system should support up to 1 petabyte of storage for multimedia content, ensuring scalability and accommodating future growth in content and user engagement.

Software design is instrumental in achieving the identified non-functional requirements for an E-Learning system as it serves as the blueprint for the entire system's architecture, functionality, and behavior. The software architecture and design choices influence the system's reliability [22]. A robust design considers

fault tolerance, error handling mechanisms, and redundancy. By incorporating these elements, software design contributes to a resilient system that can maintain stability and reliability even in the face of unexpected challenges.

Apart from that an optimized design, considering factors like load balancing and efficient data retrieval, directly influences the system's performance. Well-designed software ensures that operations are executed swiftly, meeting performance requirements.

3 ARCHITECTURE AND SOFTWARE DESIGN

3.1 Overview of design technologies at different stages of building software system

Building a software system is a multifaceted endeavor that demands a systematic approach, integrating various design technologies at each stage of the development life cycle. This section aims to provide an in-depth exploration of the diverse technologies employed throughout the different phases of software system creation. From the embryonic stages of requirements analysis to the final deployment and maintenance, the utilization of specialized design technologies is instrumental in shaping the architecture, functionality, and overall success of a software project.

The software development life cycle unfolds as a journey of conceptualization, design, implementation, testing, and deployment. At the heart of this journey lies the strategic application of design technologies, each tailored to address specific challenges and requirements unique to its stage. These technologies not only streamline the development process but also contribute to the creation of software systems that are scalable, maintainable, and aligned with user expectations.

The exploration begins with the intricate process of requirements analysis, where technologies such as Use Case Diagrams, Flowcharts, and UML empower developers and stakeholders to articulate and comprehend the intricate details of system functionality. As the journey progresses to the system architecture design, Architectural Patterns, Design Patterns, and advanced modeling tools come to the forefront, allowing for the blueprinting of a robust and scalable software foundation [23-24].

The section then delves into the realm of database design [25], showcasing

technologies like Entity-Relationship Diagrams and DBMS, pivotal in structuring and managing the data backbone of the software system. The selection of a Database Management System (DBMS), such as Oracle, MySQL [26], or PostgreSQL [27], is a critical decision influencing factors such as data retrieval speed, scalability, and security. Data modeling, including techniques like normalization, plays a crucial role in creating a conceptual representation of data entities, attributes, and relationships, forming the foundation for a well-structured database.

Moving forward, the focus shifts to the user interface design, exploring UI/UX design tools and frontend development technologies essential for crafting visually appealing and user-centric interfaces.

Backend development takes center stage with the utilization of programming languages and frameworks, setting the groundwork for the system's logic and functionality. Prototyping tools and methodologies facilitate iterative design improvements, providing a tangible representation of the evolving software system.

Testing and quality assurance introduce a plethora of technologies, from testing frameworks to continuous integration tools, ensuring the reliability and performance of the developed software. The deployment phase brings containerization technologies and deployment pipelines into play, enabling the seamless release and update of the software system.

In the upcoming sections, I will delve deeper into the intricate landscape of database design technologies. My exploration will scrutinize and compare various tools, methodologies, and frameworks, shedding light on the nuanced choices made during the design process. Join me on this scholarly expedition as we navigate the technical intricacies, evaluating the merits and nuances of different technologies to inform a comprehensive understanding of database design in modern software systems.

3.2 Database design technologies

Data has become one of the biggest assets for businesses. It's no longer just about collecting data but also about utilizing it effectively to drive business growth.

Data is the lifeblood of any successful organization and a modern fuel for digital transformation. But what many organizations don't often realize is that data also needs an organized system in order to be useful and effective. This system encapsulates the principles of database design and architecture - a crucial piece of technology that stores, structures, and organizes your corporate data into something manageable. Without a solid foundation for your tech setup, it can be difficult to access and use information effectively, leaving you lagging behind the competition and struggling just to keep up with the times. That's why understanding database design and architecture has become so important: without it, companies may find their operations hindered by inadequate storage solutions or bogged down by sluggish retrievals and analytics processes due to inefficient coding practices.

This relationship between the data elements is also commonly referred to as database schema. A database schema is a structural plan defining how data is organized within a database. It includes tables, which represent entities, columns specifying attributes, relationships between tables, constraints ensuring data integrity, and indexes for efficient data retrieval. Database schema not only tells us about the dependencies between data points but also helps to identify suitable data types and storage processes for said data. A well-designed schema is essential for effective data management, ensuring organization, consistency, and optimal performance.

Database design is essential for the success of any Database Management System (DBMS). Careful thought and consideration when designing a database can be rewarded with efficient performance, scalability, and data accuracy –

while neglecting to plan ahead properly might cause lags in your query performance and lead to other problems such as redundant data storage and data inconsistency.

3.2.1 Technologies related to Database Design

1. Database paradigm selection. It is a crucial decision-making process in database design where the appropriate database model is chosen based on the nature of the data and the requirements of the system.

This selection process involves evaluating the merits and characteristics of two primary paradigms: SQL (Structured Query Language) databases and NoSQL (Not Only SQL) databases.

2. Database Management Systems (DBMS). At the core of database design is the utilization of Database Management Systems (DBMS) (see fig. 3.1). DBMS software serves as the backbone for creating, managing, and interacting with databases. Diverse options in the market, including Oracle, MySQL, and PostgreSQL, offer distinct features and functionalities, catering to varying project requirements.

The choice of DBMS is a critical decision in the database design process, influencing factors such as data retrieval speed, scalability, and security.

3. Data modeling. Integral to the database design process is data modeling (see fig. 3.2). This involves the creation of a conceptual representation of data entities, attributes, and relationships. By providing a visual abstraction of the underlying data structure, data modeling acts as a foundation for designing an efficient and logically organized database.

Common tools used for data modeling include ER (Entity-Relationship) diagrams and UML (Unified Modeling Language), enabling developers to conceptualize and communicate the structure of the database.

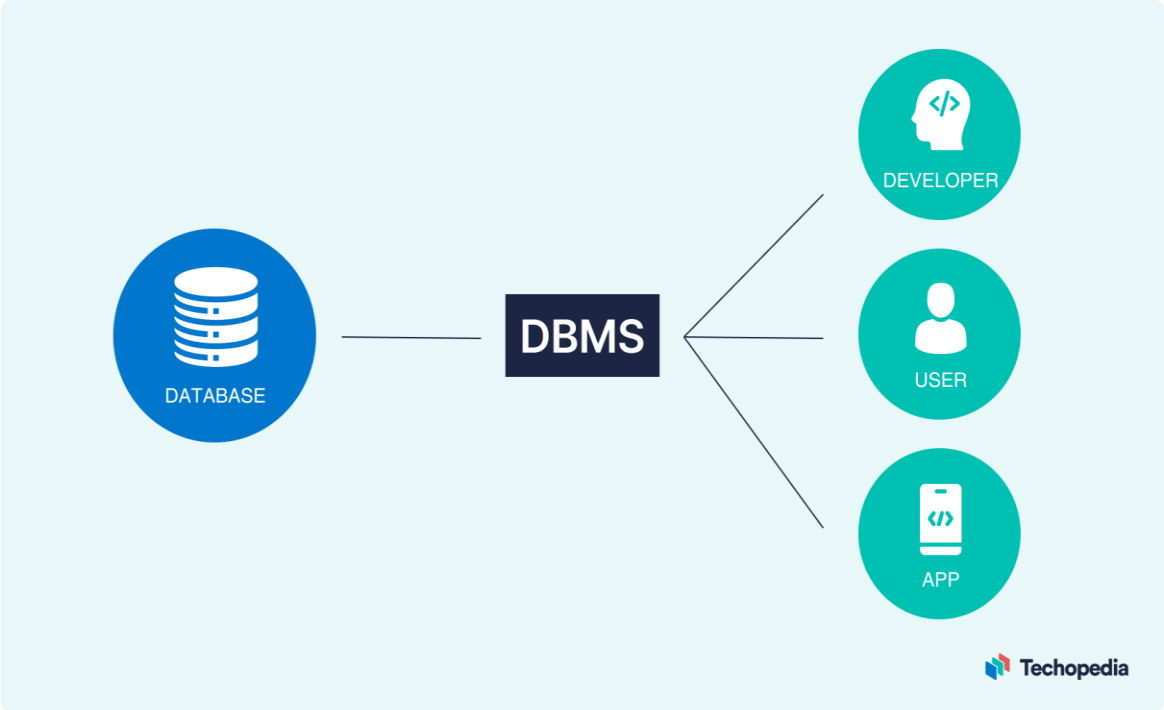

Figure 3.1 – Database Management Systems

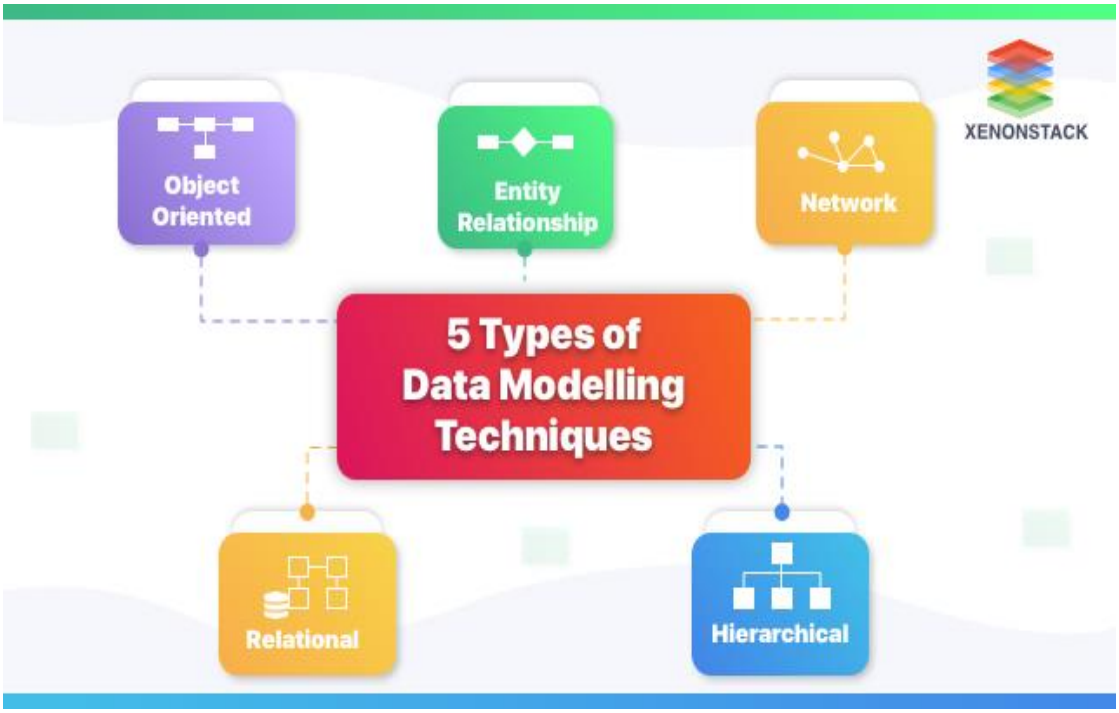

Figure 3.2 – Types of Data Modeling

4. Data normalization. Data normalization (see fig. 3.3) is a crucial technique employed in database design to enhance data integrity and eliminate redundancy. By organizing data into well-structured tables, data normalization minimizes the risk of inconsistencies and anomalies in the database. Through a series of normalization forms (1NF to 5NF), designers systematically refine the database schema, ensuring that each piece of information is stored in the most efficient and logical manner.

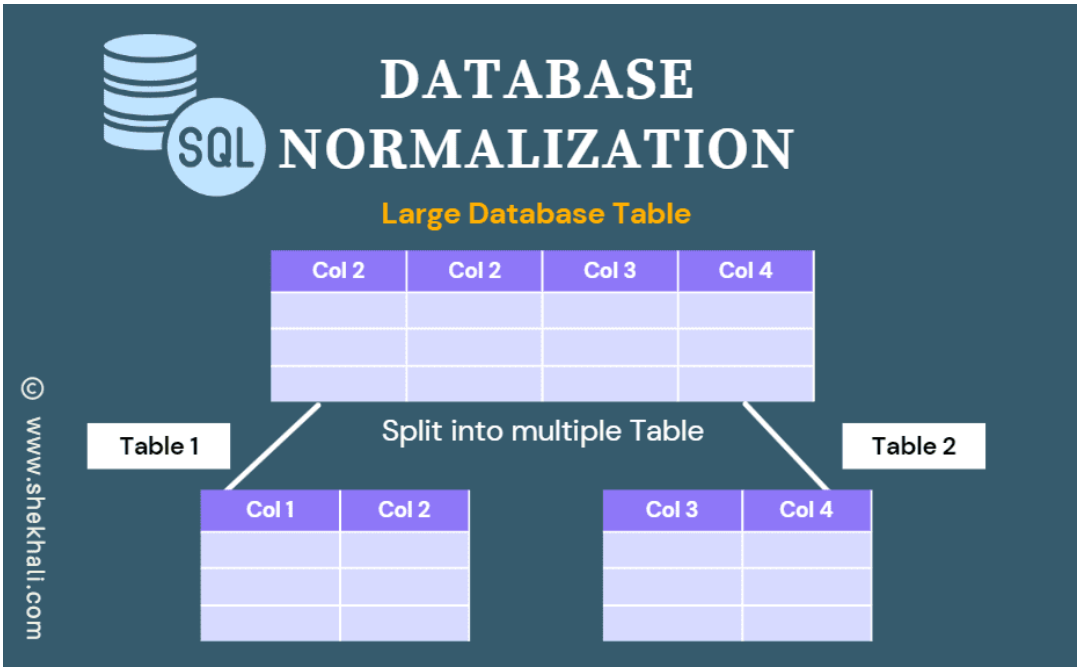

Figure 3.3 – Database normalization

5. Data migration. Data migration involves the seamless transfer of data from one database or system to another (see fig. 3.4). This process is critical during system upgrades, platform transitions, or when consolidating data from various sources. Database design principles play a pivotal role in data migration, as a well-designed database facilitates the smooth transfer of information without compromising its integrity. Meticulous planning and adherence to design standards are essential to ensure a successful data migration process.

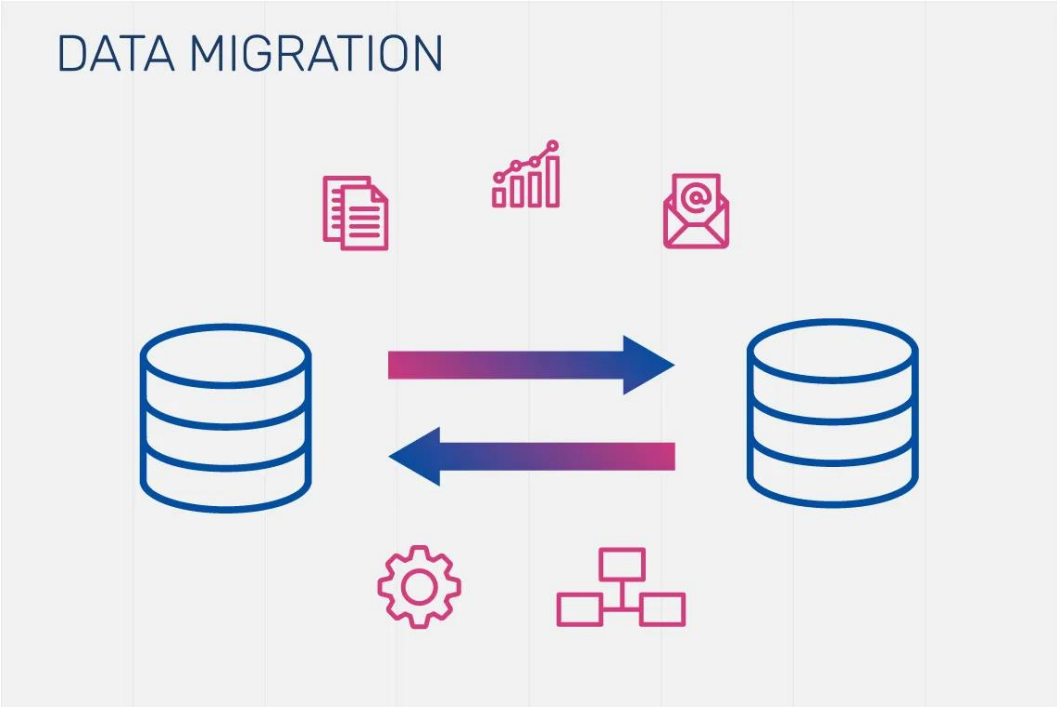

Figure 3.4 – Data migration

In the realm of database design, these technologies and terms form the cornerstone of a robust and efficient database system [28]. The judicious selection database paradigm and application of DBMS, coupled with effective data modeling, normalization techniques, and careful consideration during data migration, collectively contribute to the creation of a well-organized and high-performance database architecture.

As I embark on a more profound exploration, the focus now shifts towards an in-depth comparison of potential database paradigm selections. Delving into the intricacies of this decision-making process, I aim to unravel the nuanced distinctions between SQL and NoSQL databases [29–30]. In this comprehensive analysis, I will scrutinize the specific attributes, advantages, and considerations inherent in each paradigm, paving the way for a meticulous understanding of their applicability in diverse architectural and software design contexts.

3.3 Comparison of SQL and NoSQL database paradigm

There are two primary databases used for storing digital data: SQL (relational databases) and NoSQL (non-relational databases). Though both methods effectively store data, they differ in their structures, scalability, relationships, language, and support.

In the discerning domain of system design, the choice between SQL (Structured Query Language) and NoSQL (Not Only SQL) databases stands as a decisive crossroad, wielding significant influence over the foundational architecture governing an application's data management. This process of selection represents a strategic undertaking, demanding meticulous evaluation to discern the most fitting database model. It necessitates thoughtful consideration of the unique characteristics of the data and the specific requirements inherent to the system.

SQL databases are scalable vertically, meaning that you can increase the maximum load by adding further storage components like RAM or SSD. While in some cases this may mean that SQL databases are limited by the resources available on the server, cloud-based storage and other technologies can provide more scalability with SQL.

NoSQL databases are non-relational databases that store data in a manner other than the tabular relations used within SQL databases. While SQL databases are best used for structured data, NoSQL databases are suitable for structured, semi-structured, and unstructured data. As a result, NoSQL databases don't follow a rigid schema but instead have more flexible structures to accommodate their data-types. Furthermore, instead of using SQL to query the database, NoSQL databases use varying query languages (some don't even have a query language).

Structured Query Language (SQL) is a programming language that allows both technical and non-technically-minded users to query, manipulate, and change data in a relational database. Organized into columns and rows within a

table, SQL databases use a relational model that works best with well-defined structured data, such as names and quantities, in which relations exist between different entities. Within a SQL database, tables are linked through "foreign keys" that form relations between different tables and fields, such as customers and orders or employees and departments.

NoSQL databases are scalable horizontally, meaning that they use multiple nodes in a cluster to handle increased workloads. This allows data architects to simply scale them by supplementing clusters with additional servers. NoSQL non-relational databases work well with unstructured data and typically possess the following properties:

– NoSQL is schema-less (no fixed data model);

– NoSQL databases have a dynamic schema for unstructured data, making integrating data in certain types of applications easier and faster;

– NoSQL uses non-tabular data models, which can be document-oriented, key-value, or graph-based. The most common NoSQL databases include MongoDB, Cassandra, HBase, Redis, Neo4j, and CouchDB.

NoSQL manages the scale and agility challenges you may face in modern applications, especially ones that handle large volumes of rapidly changing data. These demands exist across every industry vertical and application domain, including IoT, user analytics, personalization, ad tech, eCommerce, gaming, and social networks.

This intricate selection process transcends the mere technical and ventures into the realm of strategic decision-making, shaping not only the efficiency of data management but also the effectiveness of content creation, delivery, and overall system responsiveness.

The intricacies of the data serve as the guiding compass in this meticulous exploration. SQL databases, characterized by their structured and tabular nature, embody a legacy of relational fidelity and transactional precision. They excel in managing well-defined, organized data structures, rendering them ideal for

systems where consistency and integrity are paramount.

Conversely, NoSQL databases embody a contemporary avant-garde, embracing the flexibility inherent in unstructured and semi-structured data. Their dynamic capabilities cater to the evolving nature of modern datasets, positioning them as essential components for systems demanding adaptability, scalability, and rapid responsiveness.

As architects and developers stand at this critical juncture, they embark on a discerning journey, navigating a complex web of considerations that delineate the very essence of their application. It is a quest for harmony, seeking equilibrium between data model flexibility and the scalability imperatives of the system. This quest necessitates a profound understanding of development speed and agility, where the nuances of a chosen query language and API design play a pivotal role.

The selection becomes an orchestrated melody resonating through the entire development process. SQL databases, with their standardized query language, provide a familiar symphony of data manipulation. In contrast, NoSQL databases compose a diverse array of languages and APIs, each tailored to the unique rhythms of their specific data models.

Yet, this is not a purely technological odyssey. It is a pursuit of consistency and transactional finesse, a striving for data integrity that materializes in the ACID properties characterizing SQL databases [31–32]. Concurrently, it involves a judicious balancing act within the NoSQL realm, where the CAP theorem introduces a nuanced interplay between consistency, availability, and partition tolerance.

As the dynamic nature of the data unfolds, architects peer into the future of their system, contemplating the security considerations that will fortify its digital fortifications. SQL databases, with their robust security features, stand as sentinels of access controls, authentication mechanisms, and encryption, ensuring the sanctity of sensitive information.

Main differences between NoSQL and SQL [33].

At a high level, NoSQL and SQL databases have many similarities. In addition to supporting data storage and queries, they both also allow one to retrieve, update, and delete stored data. However, under the surface lie some significant differences that affect NoSQL versus SQL performance, scalability, and flexibility [34].

Here are some of the main differences between SQL versus NoSQL databases: Structure, Scalability, Language, Support.

1. Structure. SQL databases are table based, while NoSQL databases can be document-oriented, key-value pairs, or graph structures (see fig. 3.5). In a NoSQL database, a document can contain key value pairs, which can then be ordered and nested.

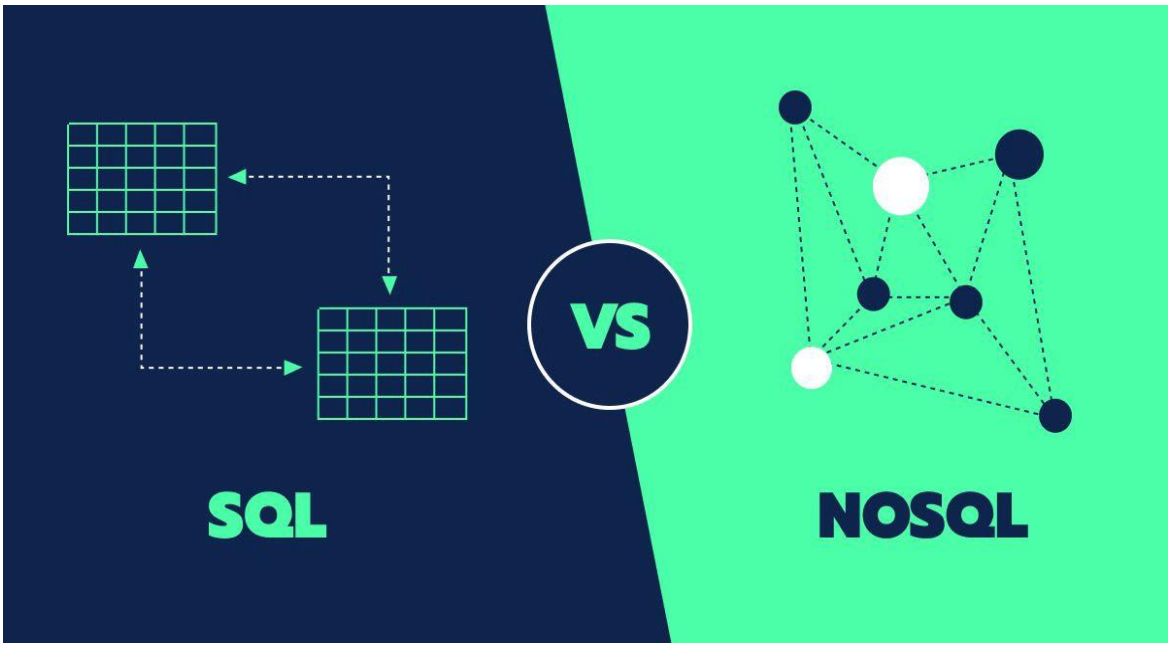

Figure 3.5 – SQL/NoSQL data structure

2. Scalability. SQL databases scale vertically, usually on a single server, and require users to increase physical hardware to increase their storage capacities. In effect, while cloud-storage options are available, SQL databases

can be prohibitively expensive for businesses when dealing with vast amounts of big data.

NoSQL databases offer horizontal scalability, meaning that more servers simply need to be added to increase their data load. This means that NoSQL databases are better for modern cloud-based infrastructures, which offer distributed resources.

3. Language. SQL databases use SQL (Structured Query Language). NoSQL databases use JSON (JavaScript Object Notation), XML, YAML, or binary schema, facilitating unstructured data. SQL has a fixed-defined schema, while NoSQL databases are more flexible.

4. Support. SQL is a popular standard language that is well supported by many different database systems, while NoSQL has varying levels of support in various database systems [35]. Regarding support, you'll generally find that more help is available for SQL databases than NoSQL. This is because SQL is a more established technology and thus has many more users and developers who can help you with your problems. In contrast, NoSQL is still relatively new, with less help available on forums or through the community. Your support options may be limited if you run into difficulties using it. Below I want to share some Pros and cons of SQL.

SQL is the lingua franca of data. It's the language you'll use most to query databases and move structured data between traditional applications. It's a powerful language that can help you do many data-related things but also has some downsides. Here are some pros and cons of using SQL for data storage and retrieval.

Pros of SQL:

a) widely understood and supported. SQL enjoys widespread adoption, and most developers are well-versed in its syntax and usage. This ubiquity makes it a universal language for database management and ensures that a large pool of

developers can work with SQL-based systems;

b) useful for simple aggregations over large datasets, such as calculating averages. SQL is particularly effective for performing simple aggregations over large datasets. Operations like calculating averages, sums, and other statistical analyses can be expressed succinctly in SQL queries, making it a powerful tool for data analysis;

c) useful for simple ETL (Extract, Transform, Load) jobs. SQL is highly valuable for setting up simple ETL jobs, especially when dealing with relational databases as both input and output sources. Its ability to transform and manipulate data within a relational framework simplifies the process of moving data between systems;

d) well-documented and easy to learn. SQL (see fig. 3.6) is well-documented, and its syntax is straightforward, making it easy for developers to learn. The simplicity and consistency of SQL contribute to its accessibility, allowing both beginners and experienced developers to grasp its concepts quickly

Figure 3.6 – SQL database

Cons of SQL:

a) performance challenges with substantial data sets. SQL performance can be suboptimal when dealing with substantial data sets, particularly during operations that require multiple passes over the data, such as complex joins. The need for iterative processing can lead to slower query execution times;

b) complicated debugging process. Debugging SQL queries can be challenging due to the limited and sometimes cryptic error messages provided by database systems. Identifying and rectifying errors may require extensive testing and analysis, leading to a more complicated debugging process.

c) verbose syntax compared to programming languages. The syntax of SQL tends to be more verbose compared to some programming languages like Python or R. This verbosity can make it harder to write complex transformations as scripts or functions. Expressing certain operations may require more lines of code, potentially leading to increased complexity and reduced readability [36].

In conclusion, while SQL is a powerful and widely used language for managing relational databases, it does come with certain drawbacks.

The performance issues on substantial data sets, debugging complexities, and the verbose syntax in comparison to some programming languages are factors that developers and database administrators need to carefully consider when choosing SQL for specific tasks or projects.

Understanding these pros and cons is crucial in making informed decisions about the selection of SQL databases, ensuring alignment with the specific needs and objectives of the project.

Pros and cons of selection of NoSQL databases. A significant benefit of NoSQL is that you don't have to define a schema upfront (or ever).

This makes it easy to add new columns without dealing with all the issues involved in altering a vast table with lots of data already in it.

It also means that if your queries don't require SQL, you can avoid the overhead of parsing and compiling SQL statements, modeling, and storing, providing an enormous performance boost when dealing with large amounts of data.

However, NoSQL is less mature than SQL. Here’s a look at NoSQL's pros and cons.

Pros of NoSQL:

a) flexible schema. NoSQL databases (see fig.3.7), in contrast, are schema-less or have a flexible schema. They can adapt to unstructured or semi-structured data, allowing for a more dynamic representation of data entities and relationships;

b) usable on distributed infrastructure platforms. NoSQL databases are designed to be used on distributed infrastructure platforms. This means they can efficiently scale horizontally by distributing data across multiple servers or nodes. This scalability is particularly beneficial for handling large volumes of data and accommodating growing workloads;

c) low-cost infrastructure. The distributed nature of NoSQL databases enables horizontal scalability, allowing organizations to scale their databases by adding more commodity hardware. This approach is often more cost-effective than vertically scaling a single, powerful server, as it leverages a larger number of more affordable machines;

d) high availability and throughput. NoSQL databases often provide high availability through features like data replication and automatic sharding. Data is duplicated across multiple nodes, ensuring that if one node fails, another can seamlessly take over. Additionally, these databases often have mechanisms for efficient load balancing, ensuring optimal performance and throughput.

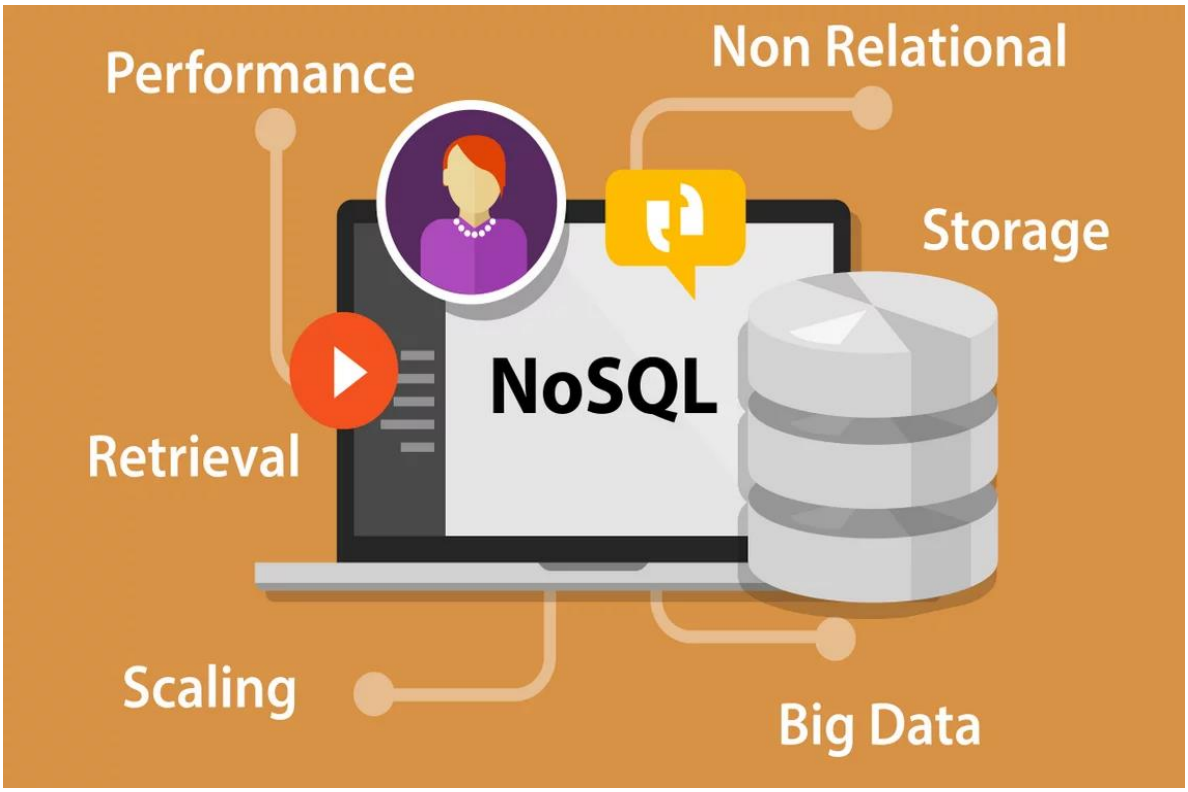

Figure 3.7 – NoSQL database

Cons of NoSQL:

a) less mature technology and difficulty in management. NoSQL databases, as a technology, are often considered less mature than traditional SQL databases. This can result in challenges related to stability, documentation, and overall ecosystem maturity. Managing and maintaining NoSQL databases might require more effort and expertise;

b) limited query capabilities. NoSQL databases, depending on the type (document-oriented, key-value stores, etc.), may have limited query capabilities compared to SQL databases. Ad-hoc querying and complex joins may not be as straightforward, potentially leading to challenges in expressing certain types of queries;

c) data inconsistency and performance challenges in complex scenarios. some NoSQL databases prioritize scalability and partition tolerance over strict consistency, leading to eventual consistency models. In scenarios where strong consistency is crucial, the eventual consistency approach may result in data inconsistencies. Additionally, certain complex scenarios, especially those requiring complex transactions or multi-document transactions, might experience performance challenges.

The flexibility, scalability, and cost-effectiveness of NoSQL databases make them a preferred choice for certain use cases, especially in scenarios where rapid growth, dynamic data structures, and high availability are paramount. It's essential to carefully consider the specific requirements of a project and the nature of the data to determine whether a NoSQL database is the most suitable option.

In conclusion, the selection between SQL and NoSQL databases hinges on the specific needs and objectives of the project. There is no one-size-fits-all solution, and the optimal choice depends on factors such as data structure, scalability requirements, development speed, and the nature of the application. Striking a balance between the structured reliability of SQL databases and the dynamic flexibility of NoSQL databases is key to achieving a well-aligned and high-performance database architecture. Whether navigating the relational terrain of SQL or embracing the flexible horizons of NoSQL, a thorough understanding of the pros and cons empowers decision-makers to architect robust and efficient database solutions tailored to the demands of their unique projects.

4 DESCRIPTION OF SELECTED INFORMATION TECHNOLOGY FOR DESIGNING E-LEARNING SOFTWARE SYSTEM

4.1 Solution for database design technology

Aligned with the thorough evaluation of advantages and drawbacks delineated in the previous section, combined with the specific requisites defined for an E-Learning software system, the decision to select a SQL database becomes apparent. This strategic decision is grounded in a deep understanding of the nuances inherent in various database paradigms, coupled with a meticulous consideration of the distinctive requirements characteristic of E-Learning platforms. The comprehensive understanding I have cultivated regarding the advantages and drawbacks inherent in various database paradigms has played a pivotal role in steering this decision-making process. By acknowledging the specific requisites of E-Learning platforms, this decision seeks to lay the foundation for an efficient, scalable, and adaptable system that not only meets but exceeds the expectations of modern educational environments. Each consideration, from the intricacies of structured querying to the subtleties of data normalization, has been weighed and measured against the canvas of E-Learning demands.

Educational technology, with its multifaceted demands and diverse user interactions, necessitates a database solution that can seamlessly blend reliability, scalability, and efficient data management. The following elucidates the rationale for opting for a SQL database in the context of crafting an E-Learning software system.

1. Structured learning environments demand structured data.

In the realm of E-Learning, where structured content delivery and organized user interactions are foundational, SQL databases align harmoniously.

The tabular structure with predefined schemas ensures that educational content, user profiles, and complex relationships are maintained in a systematic and well-defined manner.

2. Empowering educators with complex querying capabilities.

E-Learning systems require robust querying capabilities for educators to derive meaningful insights into student performance, course effectiveness, and overall system usage. SQL's standardized query language empowers educators with the tools to perform complex queries effortlessly, fostering data-driven decision-making in the educational ecosystem.

3. Ensuring reliability through ACID properties.

The inherent ACID properties (see fig 4.1) of SQL databases contribute to the reliability and consistency of data transactions. In an educational setting where accurate record-keeping and dependable transaction management are non-negotiable, SQL databases ensure that each operation is executed with precision, maintaining the integrity of critical educational data.

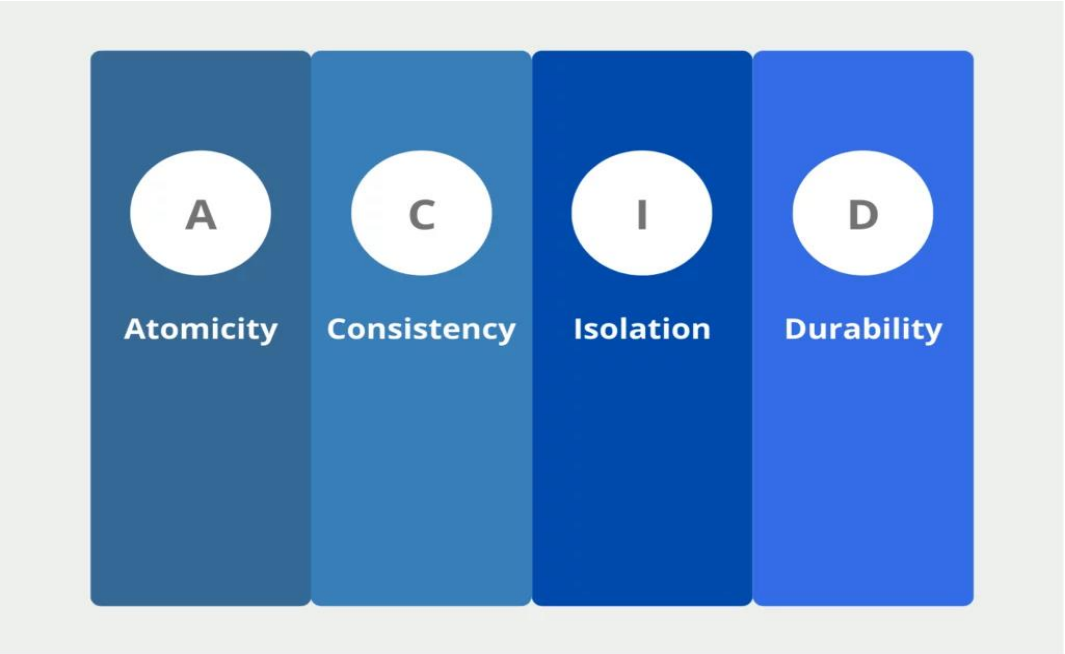

Figure 4.1 – ACID properties

4. Safeguarding sensitive educational data with robust security.

As guardians of sensitive user information, E-Learning systems require a robust security framework. SQL databases provide a comprehensive suite of security features, including stringent access controls, authentication mechanisms, and encryption. These measures are indispensable for creating a secure environment, ensuring the confidentiality and privacy of educational data.

5. Scalability to accommodate growing educational demands.

The scalability of an E-Learning system is paramount, especially considering potential growth in user enrollment and content volume. SQL databases, with their ability to scale vertically, provide an adaptable solution that can gracefully accommodate increasing demands without compromising on system performance, guaranteeing a seamless learning experience.

6. Optimized data management for efficiency.

Data normalization, a fundamental practice in SQL database design, optimizes data storage by eliminating redundancy. In the context of E-Learning, where diverse entities interconnect, normalized data structures enhance efficiency. This streamlined approach ensures faster query execution and responsiveness, enhancing the overall user experience.

7. Seamless integration within the educational technology ecosystem.

SQL databases, with their widespread adoption, seamlessly integrate with various educational technology tools and frameworks. This compatibility fosters a cohesive ecosystem where the E-Learning system can effortlessly communicate with existing tools, platforms, and services, creating a unified and interoperable educational technology landscape.

8. Transaction support.

The transactional support provided by SQL databases ensures the reliability of complex operations. In the context of e-learning systems, where user interactions, content updates, and analytics are intricate processes, robust transaction support is imperative for system stability.

As we delve into the intricacies of designing an E-Learning software system with SQL at its core, this decision represents more than a technical choice; it symbolizes a commitment to excellence, reliability, and user-centric educational experiences. The synthesis of technology and pedagogy unfolds in the chapters ahead, where the foundations laid in the comparative exploration of database paradigms pave the way for a tailored, robust, and efficient E-Learning solution.

4.2 Developed database for the E-Learning system

As a result of exploration and careful consideration of the merits and demerits of SQL databases, a pivotal decision has emerged. This decision manifests not only as a selection of a technology stack but as a strategic commitment to shape the foundation of a basic E-Learning system.

Through a meticulous process of comparison and weighing the pros and cons, I have ventured into the realm of database design, crafting a framework that encapsulates essential entities and their intricate relationships.

This endeavor goes beyond a mere technical choice; it signifies the orchestration of a foundational structure that aspires to redefine the landscape of educational technology.

Each entity within the database schema is a carefully curated building block, designed to enhance the functionality, reliability, and scalability of the envisioned E-Learning system. As this exploration transpires, it lays the groundwork for a software solution that not only meets functional requirements but also aligns with the dynamic expectations of modern educational environments.

This database schema aspires to transcend mere functionality, aspiring to

be a catalyst for excellence in user experience and system performance. As the journey unfolds, this database design stands as a testament to the commitment to create a robust and efficient E-Learning system that paves the way for future advancements in the field.

The entities and relationships intricately woven into the database fabric echo the culmination of thoughtful decisions, guided by a deep understanding of the nuances of SQL databases.

The ER diagram of the database (see fig. 4.2) illustrates the entities present in the system and the relationships between them. All entities are represented as tables, each with its set of fields.

The database contains all the information necessary for the full functioning of the system. Thus, the tables store data about the user, their role in the system, and information about the organization of the system as a whole. The main table can be considered the Users table, which will store personal information about users of the system. All users are categorized and have different privileges.

As displayed on the ER diagram there are 5 main entities: Course, Module, User, Admin, Progress.

These entities form the foundation of the database schema, representing key components in the design of the E-Learning system. Each entity serves a specific role in ensuring the system's functionality and efficiency:

1. Course:

– description: represents a distinct course within the E-Learning system.

– role: serves as the core unit around which the educational content is organized;

– attributes: includes properties like CourseID, Level, Title, Description, Duration, and any other relevant information.

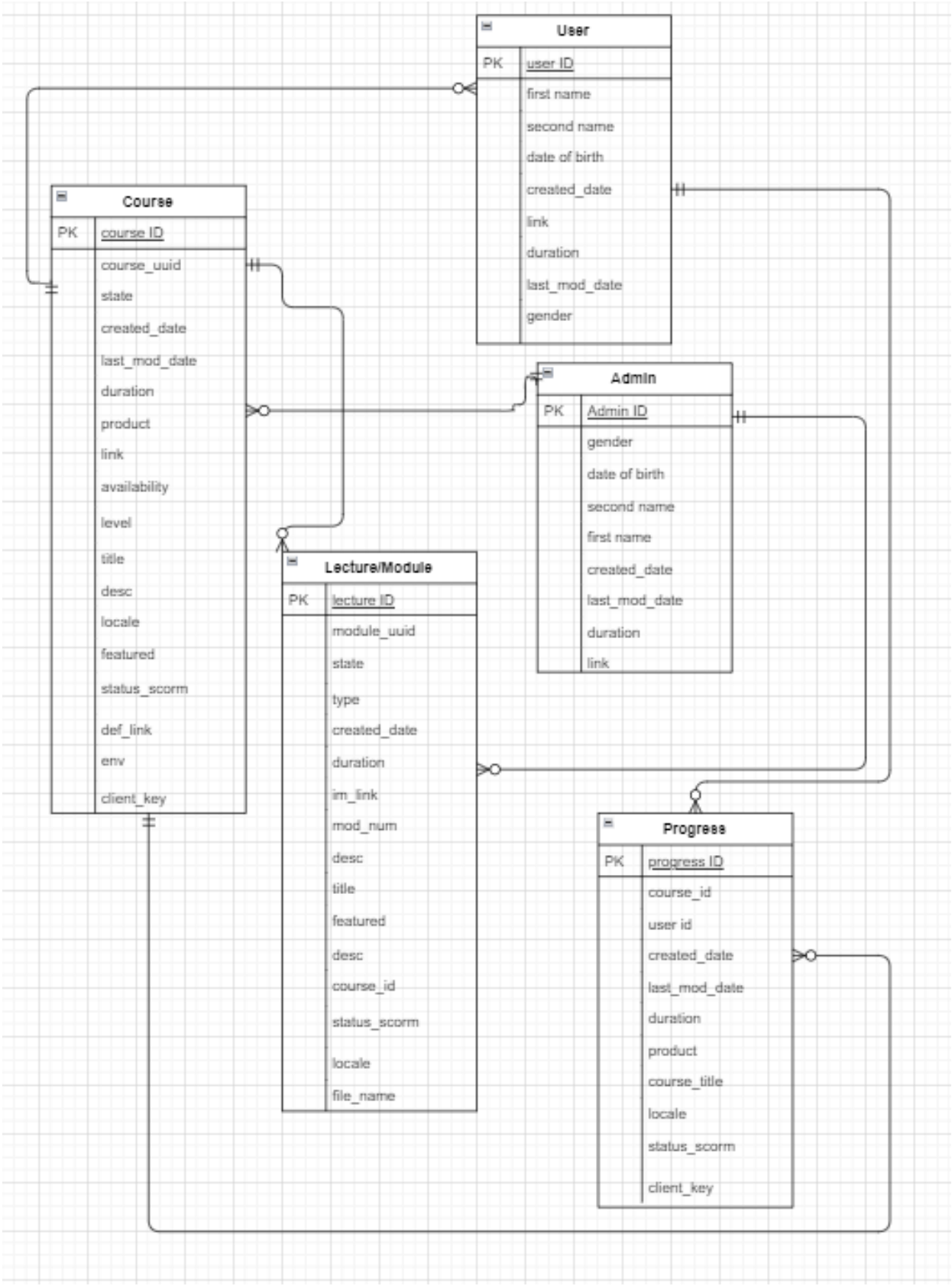

Figure 4.2 – ER diagram of the E-Learning system

2. Module:

– description: represents a module or unit of content within a course;

– role: acts as a subdivision of a course, focusing on a specific topic or learning objective;

– attributes: includes ModuleID, Title, Type, Duration, CourseID (foreign key) for showing relationship with related courses and other details specific to each module.

3. User:

– description: represents an individual user registered within the E-Learning system;

– role: encompasses both learners and instructors using the platform;

– attributes: typically includes properties like UserID, FirstName, SecondName, Email, and other user-specific details.

4. Admin:

– description: represents an administrative user with elevated privileges;

– role: manages and oversees the overall functioning of the E-Learning system;

– attributes: includes AdminID, FirstName, SecondName, Email, and additional attributes based on administrative needs.

5. Progress:

– description: represents the progress of a user within a course or module.

– role: tracks and records the user's advancement through the learning content;

– attributes: includes ProgressID, UserID (foreign key), CourseID/ModuleID (foreign key), CompletionStatus, and other relevant

progress-related details.

These entities collectively form a relational structure, allowing for efficient organization and retrieval of information within the E-Learning system. Users, both learners and administrators, interact with courses and modules, and the progress entity ensures that the system can track and manage the users' learning journey.

CONCLUSIONS

In the culmination of this master's work, a comprehensive exploration unfolded, delving into various facets of software system design for distance learning. The journey commenced with an in-depth analysis of existing E-Learning systems, identifying crucial aspects that impact the efficacy of such platforms. The requirements for the software system were also formulated. As the comprehensive analysis progressed, the focus shifted beyond the functional aspects of the software system to the critical realm of non-functional requirements. These non-functional requirements play a pivotal role in defining the quantitative and qualitative performance indicators that are integral for the successful operation of the system.

As requirements for an ideal E-Learning system crystallized, the focus shifted to the intricate realm of database design.

A meticulous comparison between SQL and NoSQL databases unfolded, weighed against the specific needs of the envisioned system. The strategic decision to embrace SQL as the database technology was substantiated by its well-established principles, structured query capabilities, and robust data integrity mechanisms. This choice represents not just a technical decision but a deliberate commitment to ensuring efficiency, reliability, and scalability.

The software development life cycle, explored in its entirety, highlighted the strategic application of design technologies at each stage. From requirements analysis to deployment, each phase was underscored by the integration of specialized design technologies, contributing to scalable, maintainable, and user-aligned software solutions.

Non-functional requirements emerged as crucial parameters, defining the quantitative and qualitative performance indicators essential for the system's success. The synthesis of functional and non-functional requirements laid the

foundation for a software solution that excels in operational efficiency, user satisfaction, and adaptability to evolving needs.

The selection of SQL as the database technology for the designed E-Learning system is grounded in its compatibility with the specific requirements of the educational landscape. SQL databases provide a structured and reliable framework for managing data, ensuring efficient querying and maintaining consistency, vital attributes for the dynamic and content-intensive nature of E-Learning platforms.

In summary of this master's work, the choices made, technologies explored, and insights gained contribute to the broader discourse of software architecture, with a specific focus on enhancing the landscape of educational technology. The choice of SQL for the E-Learning system is not just a technological decision but a strategic one that aligns with the unique demands of educational environments. It enhances data management, ensures efficient access to educational content, and contributes to the overall reliability and scalability of the system – a choice poised to elevate the E-Learning experience for both educators and learners.

The results of the research were published in the abstracts of the international scientific and practical conference «Computer-integrated technologies for automation of technological processes in transport and production» [18].

LIST OF RESOURCES

1. Pressman R., Maxim B. Software Engineering: A Practitioner's Approach. 2019. 705 p. URL: https://dokumen.pub/software-engineering-a-practitioners-approach-9nbsped-1260548007-9781260548006.html (date of reference: 18.11.2023).

2. Humble J., Farley D. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional. 2011. 464 p. URL: https://balka-book.com/razrabotka_programnogo_obespecheniya-366/continuous_delivery_reliable_software_releases_through_build_test_and_deployment_automation-32901 (date of reference: 18.11.2023).

3. Udacity Blog: Udacity August Newsletter: All the latest updates straight. URL: https://www.udacity.com/online-learning-for-individuals?utm_source=gsem_brand&utm_medium=ads_r&utm_campaign=12949811881_c_individuals&utm_term=123473112204&utm_keyword=%2Budacity_b&gclid=CjwKCAjwtIaVBhBkEiwAsr7-c0OKiroW27uJ8WkrZX1-RaOVwU7HebVlDXwdTofHLR-VSl2gAKdxiBoC3kcQAvD_BwE (date of reference: 06.11.2023).

4. Frederick P., Brooks Jr. The Mythical Man-Month: Essays on Software Engineering. URL: https://web.eecs.umich.edu/~weimerw/2018-481/readings/mythical-man-month.pdf (date of reference: 18.11.2023).

5. Regan. Google's Classroom is open to anyone with an urge to teach. Engadget. AOL (April 27, 2017).URL: https://www.engadget.com/2017-04-27-googles-classroom-is-open-to-anyone-with-an-urge-to-teach.html (date of reference: 06.11.2023).

6. The Daily Illini: Coursera partner to educate masses. URL: https://dailyillini.com/article/2012/07/ui-coursera-partner-to-educate-masses

(date of reference: 06.11.2023).

7. System Design in Software Development. URL: https://medium.com/the-andela-way/system-design-in-software-development-f360ce6fcbb9 (date of reference: 06.11.2023).

8. IT System Design: Tools & Technologies in System Design. URL: https://medium.com/@anant3104/it-system-design-tools-technologies-in-system-design-part-5-1e6df27f60d2 (date of reference: 06.11.2023).

9. What is system design and why is it necessary? URL: https://segwitz.com/what-is-system-design-and-why-it-is-necessary/ (date of reference: 06.11.2023).

10. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. 2017. 614 p.

11. Hunt A., Thomas D. The Pragmatic Programmer: Your Journey to Mastery. 2019. 352 p.

12. Refactoring: Improving the Design of Existing Code / Fowler M., Beck K., Brant J., Opdyke W. 1999. 464 p.

13. Beskorovainyi V. Combined method of ranking options in project decision support systems // Innovative Technologies and Scientific Solutions for Industries. 2020. No 4 (14). P. 13–20. URL: http://journals.uran.ua/itssi/article/view/ITSSI.2020.14.013 (date of reference: 17.12.2023).

14. Beskorovainyi V. V., Petryshyn L. B., Shevchenko O. Yu. Specific subset effective option in technology design decisions // Applied Aspects of Information Technology. 2020. Vol. 3. No. 1. P. 443–455. URL: https://aait.op.edu.ua/?fetch=articles&with=info&id=40 (date of reference: 18.11.2023).

15. Beskorovainyi V., Berezovskyi H. Identification of preferences in decision support systems // ECONTECHMOD. 2017. Vol. 06. №4. P. 15–20. URL: https://openarchive.nure.ua/items/9a27c447-47cc-44b0-b0c0-e6fffaaccb10 (date of reference: 18.11.2023).

16. Beskorovainyi V., Berezovskyi H. Estimating the properties of technological systems based on fuzzy sets // Innovative technologies and scientific solutions for industries. 2017. № 1 (1). С. 14–20. DOI: https://doi.org/10.30837/2522-9818.2017.1.014 (date of reference: 18.11.2023).

17. Beskorovainyi V., Kolesnyk L., Russkin V. Decision making support under conditions of incomplete consistency of expert advantages // Innovative integrated computer systems in strategic project management": Collective monograph edited by I. Linde. European University Press. Riga: ISMA, 2022. P. 16-26. URL: https://mmp-conf.org/documents/archive/monography2022.pdf (date of reference: 18.11.2023).

18. Laktionova A. O., Bezkorovainyi V. V. Rozrobka komponentiv informatsiinoi tekhnolohii E-Learning [Development of E-learning information technology components] // Kompiuterno-intehrovani tekhnolohii avtomatyzatsii tekhnolohichnykh protsesiv na transporti ta u vyrobnytstvi. Materialy vseukr. nauk.-prakt. konf. zdobuvachiv vyshchoi osvity i molodykh uchenykh. Kharkiv, KhNADU [Computer-integrated technologies for automation of technological processes in transport and production. All-Ukrainian materials science and practice conference students of higher education and young scientists. Kharkiv, KhNADU], 2023. P. 202–206 [in Ukrainian] URL: https://dspace.khadi.kharkov.ua/items/6f1cfc19-697c-4d38-9516-d9a52b7d4b47 (date of reference: 18.12.2023).

19. Beaulieu A. Learning SQL: Generate, Manipulate, and Retrieve Data. 2020. 380 p.

20. Sadalage P. J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. 2012. 234 p. URL: https://bigdata-ir.com/wp-content/uploads/2017/04/NoSQL-Distilled.pdf (date of reference: 18.11.2023).

21. McConnell S. Code Complete: A Practical Handbook of Software Construction. 2004. 952 p.

22. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall. 2009. 464 p.

23. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. Addison-Wesley Professional. 1995. 396 p.

24. Freeman E., Robson E. Head First Design Patterns. 2020. 672 p.

25. Hernandez M. J. Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design. 2003. 640 p.

26. Schwartz B., Zaitsev P., Tkachenko V. High Performance MySQL: Optimization, Backups, and Replication. 2012. 826 p. URL: https://www.academia.edu/27289779/High_Performance_MySQL_Optimization_Backups_and_Replication (date of reference: 06.11.2023).

27. PostgreSQL: Up and Running. A Practical Guide to the Advanced Open Source Database. 3rd Edition. URL: https://www.amazon.com/PostgreSQL-Running-Practical-Advanced-Database/dp/1491963417 (date of reference: 06.11.2023).

28. Date C. J. Introduction to Database Systems". 8-th ed. Addison-Wesley Professional. 1024 p.

29. Redmond E., Wilson J. R. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement. Pragmatic Bookshelf 2012. 333 p.

30. Date C. J., Darwen H., Lorentzos N. A. A Guide to the SQL Standard. Addison Wesley Publishing Company. 2006. 414 p.

31. 5 Foundational Design Patterns for Data Modelling. URL: https://martinterhaak.medium.com/data-modeling-design-patterns-part-1-3fbd45a8392 (date of reference: 06.11.2023).

32. ACID Properties in DBMS URL: https://www.geeksforgeeks.org/acid-properties-in-dbms/ (date of reference: 06.11.2023).

33. SQL vs. NoSQL Databases: What’s the Difference? URL:

https://www.ibm.com/blog/sql-vs-nosql/ (date of reference: 06.11.2023).

34. Winand M. SQL Performance Explained. 2012. 204 p.

35. Garcia-Molina H, Ullman J. D., Widom J. Database Systems: The Complete Book. 2009. 1248 p.

36. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. 2003. 560 p.