

ЗМІСТ

Вступ.....	5
1 Тематичний огляд.....	9
1.1 Принципи електронного навчання та проектування	9
1.2 Система електронного навчання	12
1.3 Процес електронного навчання на основі стандарту SCORM	14
1.4 Модель керування процесом електронного навчання SCORM.....	19
1.5 Математична модель управління процесом електронного навчання	21
1.6 Огляд існуючих систем та програмних рішень.....	25
1.7 Висновки до першого розділу.....	30
2 Дослідження моделі та методу тестування	31
2.1 Постановка задачі.....	31
2.2 Аналіз переваг та недоліків методів тестування знань слухачів	33
2.3 Мета-модель оцінювального простору для тренінгу	35
2.4 Алгоритм для генерації та оцінки тестів для тренінгу.....	37
2.5 Математична модель на основі цілочисельного програмування для генерації та оцінки тестів професійного тренінгу	39
2.6 Верифікація алгоритму для генерації та оцінки тестів професійного тренінгу.....	41
2.7 Висновки до другого розділу	45
3 Визначення архітектури системи	46
3.1 Дослідження загальної архітектури системи	46
3.2 Дослідження динамічної архітектури системи	49
3.3 Висновки до третього розділу.....	53
4 Реалізація та верифікація.....	54
4.1 Проектування серверної та клієнтської частини системи	54
4.2 Проектування та розробка діаграми класів	58
4.3 Розробка бази даних.....	59
4.4 Верифікація розробленої моделі генерації та оцінки тестів на практиці	63
4.5 Порівняння та аналіз результатів прототипу	65

4.6 Висновки до четвертого розділу	67
Висновки	68
Список використаної літератури	69

ВСТУП

Оцінювання є невід'ємним інструментом в процесі навчання. Знання слухачів можна оцінити з урахуванням теоретичних знань і практичних навичок.

Найбільш поширеним способом теоретичної оцінки успішності слухачів є тестування. Тести як інструмент перевірки знань і подальшої оцінки, не тільки використовуються у традиційному навчанні, а й у різних формах електронного навчання. Нові завдання електронного навчання вимагають впровадження сучасних технологій в інструменти бізнес-аналізу для підвищення якості продукту та задоволеності користувачів.

В даний час персоналізовані послуги навчання є ключовим моментом у галузі онлайн-навчання, оскільки не існує єдиного плану навчання, що підходить для всіх слухачів. Однак традиційні системи навчання ігнорують ці вимоги до сервісів та надають усім слухачів однаковий навчальний контент. Такий підхід може бути неефективним для слухачів з різним рівнем підготовки та здібностей. Щоб розробити адаптивний навчальний контент, необхідно забезпечити можливість доставки навчального контенту відповідно до потреб конкретного слухача. За допомогою LMS наставники можуть створювати та інтегрувати матеріали курсу, формулювати цілі навчання, узгоджувати зміст та оцінки, відстежувати прогрес у навчанні та створювати індивідуальні тести для слухачів. LMS дозволяє доносити цілі навчання та організовувати навчальні графіки. Перевага LMS у тому, що вона доставляє навчальний контент та інструменти безпосередньо слухачам, а оцінка може бути автоматизована. Вона також може охоплювати маргіналізовані групи за допомогою спеціальних налаштувань. Такі системи мають вбудовані функції, що настроюються, включаючи оцінку і відстеження. Отже, слухачі можуть відстежити у реальному часі свій прогрес, а наставники можуть контролювати й повідомляти про ефективність навчання. До 2021 року майже всі великі навчальні заклади перейшли на електронне навчання. З розвитком соціальних мереж віртуальні навчальні середовища стали більш інтерактивними, спільними і розмовними. В даний час e-learning узятий на

озброєння різними компаніями для інформування та навчання як співробітників компанії, так і клієнтів.

Дослідження моделей і методів аналізу властивостей запровадження LMS підтверджують доцільність відповідних етапів у проектуванні, конструюванні, верифікації, налагодженні систем електронного навчання, що функціонують в середовищі LMS.

Завдання дослідження і побудови формальних моделей і методів для аналізу структури поведінки LMS з її реалізацією представляють інтерес з точки зору розробки сучасних LSM, є актуальними і нетривіальними.

Мета роботи - пропозиція інтелектуальної системи для створення та оцінки тестів електронного навчання з використанням цілісного програмування. Основою цієї системи є формулювання математичних моделей для генерації тестів з більшою кількістю питань, але з меншим ступенем складності, і для генерації тестів з меншою кількістю питань, але з більшим ступенем складності.

Майбутня робота зосереджена на включенні запропонованої моделі в інтерактивну програмну систему, здатну створювати та надавати зміст електронного навчання відповідно до конкретних потреб та попередніх знань слухачів. У цьому напрямку важливою областю, що визначає нові можливості та завдання для майбутніх досліджень, є розробка відповідного середовища електронного навчання для слухачів із застосуванням запропонованої моделі. Зрештою, слід зазначити, що перехід до персоналізованого навчання на основі попередніх знань слухачів – це процес, який, безумовно, стане майбутнім електронного навчання. В даний час через комплекс технологічних та нетехнологічних факторів цей процес складний і неефективний. З розвитком ефективних рішень у цій галузі, що підтримуються передовими технологіями, такими як штучний інтелект та аналіз великих даних, нові досягнення поступово усуватимуть технологічні проблеми і як їх наслідок - знижувати витрати та підвищувати мотивацію як творців контенту персоналізованого навчання, так і самих слухачів.

Поставлені завдання. Для досягнення мети вирішені завдання:

- дослідження специфікацій та стандартів e-learning;
- аналізу існуючих методів інтеграції майбутнього функціоналу в систему LMS;

- побудови, аналізу і моделювання функціональних моделей й методів котрі здатні створювати та надавати зміст електронного навчання відповідно до конкретних потреб та попередніх знань слухачів.;

Об'єктом дослідження є модель керування процесом електронного навчання.

Предметом дослідження є процес інтеграції запропонованих моделей електронного навчання та тестування на основі системи LMS.

Методи аналізу та проектування ґрунтуються на застосуванні загальної і булевої алгебри, теорії предикатів, контролю і діагнозу, використовуваних як математична база побудови моделей, теорій алгоритмів, цілочисельного програмування метою котрого є математична оптимізація чи доцільність використання, у якій деякі чи всі змінні мають бути цілими числами, об'єктного, компонентного програмування в якості методологічної бази побудови засобів системи електронного оцінювання.

Наукова новизна дослідження полягає в тому, що в роботі удосконалено мета-модель та метод оцінювального простору для професійного тренінгу.

Розглянуто формальну модель агрегації контенту основна структура моделі агрегації контенту складається з організацій контенту та об'єктів навчання. Організація контенту будується деревоподібною структурою елементів, а навчальні об'єкти представлені ресурсами, кожен із яких пов'язані з SCO (Sharable Content Object) чи активом, вибраним з бункера навчальних об'єктів. Більш того, ці елементи моделі агрегації контенту точно визначені пакетом SCORM. Тобто специфікація упаковки LMS (e-Learning Management System) використовує SCORM як основу для пакування та організації навчальних об'єктів. Пакет може містити більше однієї організації контенту для тих самих об'єктів навчання, що передбачає, що кілька процесів електронного навчання, що охоплюють один і той

же предмет на різних рівнях глибини або для різних аудиторій, можуть бути визначені в одному пакеті SCORM.

Практичне значення отриманих результатів полягає в запропонуванні інтелектуальної системи для проходження тренінгів та навчання, оцінки тестів електронного навчання з використанням цілісного програмування, котра базується на удосконалених моделях та стандарту SCORM та основній концепції навчання. Основою цієї системи є формулювання математичних моделей для генерації тестів з більшою кількістю питань, але з меншим ступенем складності, і для генерації тестів з меншою кількістю питань, але з більшим ступенем складності.

Пояснювальна записка складається з вступу, розділу бібліографічного аналізу електронного навчання, розділу розробки та дослідження моделей генерації та оцінювання тестів для тренінгового модулю, розділу розробки і аналізу методів інтеграції запропонованих рішень в систему, розділу реалізації системи проходження тренінгу і експериментів для моделей і методів, висновків, бібліографії в складі 23 джерел.

1 ТЕМАТИЧНИЙ ОГЛЯД

1.1 Принципи електронного навчання та проектування

Електронне навчання (англ. E-learning, скорочення від англ. Electronic Learning) – система навчання, за допомогою інформаційних, електронних технологій. Часто тлумачиться, як синонім таких понять: дистанційне навчання, навчання з застосуванням комп'ютерів, мережеве навчання, віртуальне навчання, мультимедійне навчання, мобільне навчання. Теорія електронного навчання описує когнітивні наукові принципи ефективного мультимедійного навчання з використанням електронних освітніх технологій.

До електронного навчання відносяться електронні підручники, освітні послуги та технології. Фактично електронне навчання почалося з використанням комп'ютерів в освіті. Спочатку навчання з використанням комп'ютерів чергувалось звичайними, класичними практичними заняттями. E-learning і зараз не виключає спілкування з наставником лицем до лиця.

Найвідоміші «системи» в e-learning - це системи дистанційного навчання (СДН), або системи управління навчанням (learning management system, LMS.) Термін настільки широко вживаним, що часто вживається як повний синонім, тобто «впровадити e-learning» прирівнюється до «придбати і налаштувати СДО або LSM».

Електронна освіта має ряд переваг перед традиційною: [1]

– Свобода доступу - учень може займатися практично в будь-якому місці. Далеко не всі функції електронної освіти реалізуються через інтернет. Дорослий учень може навчатися без відриву від основної роботи.

– Зниження витрат на навчання - учень несе витрати на носій інформації, але не несе на методичну літературу. Крім того, економія зростає за рахунок зарплат, які не потрібно платити педагогам, утримання навчальних закладів і так далі. Виробництво електронних навчальних матеріалів не має на увазі вирубки лісу.

– Гнучкість навчання - тривалість і послідовність вивчення матеріалів слухач вибирає сам, повністю адаптуючи весь процес навчання під свої можливості і потреби.

– Можливість розвиватись згідно з часом - користувачі електронних курсів: і наставники, і слухачі розвивають свої навички і знання відповідно до новітніх сучасних технологій і стандартів. Електронні курси також дозволяють своєчасно і оперативно оновлювати навчальні матеріали.

– Потенційно рівні можливості навчання - навчання стає незалежним від якості викладання в конкретному курсі навчання.

– Можливість визначати критерії оцінки знань - в електронному навчанні є можливість виставляти чіткі критерії, за якими оцінюються знання, отримані слухачами в процесі навчання.

Проектування та розвиток освіти (Learning design, LD) – це поступовий процес, який систематично проходить етапи аналізу, проектування, розробки, впровадження та оцінки. Інструментарій кожного етапу різниться залежно від конкретних цілей, умов та учасників, які грають роль цьому етапі. Специфікація LD - це система позначень, що дозволяє здійснювати гнучку, персоналізовану, інтерактивну освіту, засновану на різних педагогічних поглядах.

Більшість формальних стратегій проектування дизайну навчання починають міркування з мети навчання, але можна також почати з навчальної діяльності, допоміжної діяльності (зазвичай забезпечується вчителем) або середовищем. Часто багато змінних проектування вже визначено і таким чином є константами в процесі встановлені. Наприклад, у більшості ситуацій ролі заздалегідь визначені (студент, викладач, наставник, оцінювач тощо), а також глобальний розклад (наприклад, семестри). Якщо орієнтуватися на традицію передачі знань, то мається на увазі, що навчальна діяльність завжди є варіантами на тему: "засвоїти надані знання". У цьому випадку можна зосередитись на питанні, які знання та які тестові ресурси необхідно надати.

Основна концепція специфікації дизайну навчання, виражена на рисунку 1.1, вона полягає у тому, що незалежно від педагогічного підходу, людина

отримує певну роль процесі викладання-навчання, зазвичай роль учня чи співробітника. У цій ролі він або вона працює над досягненням певних результатів, виконуючи більш менш структуровану навчальну та/або допоміжну діяльність у середовищі. Середовище складається з відповідних навчальних об'єктів та послуг, які будуть використовуватись під час виконання діяльності. Яка роль отримує яку діяльність у якийсь момент процесу, визначається методом або повідомленням.

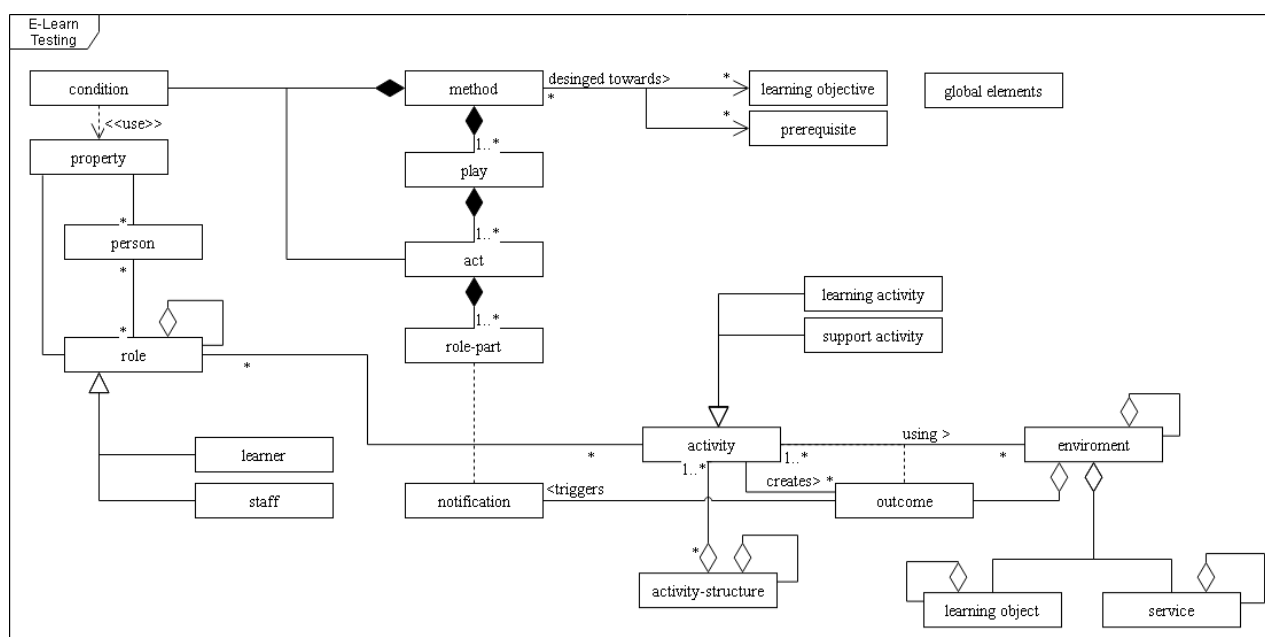


Рисунок 1.1 – Основна концепція дизайну навчання

Метод призначений для досягнення цілей навчання (специфікація результатів для учнів) та передбачає наявність певних передумов (специфікація початкового рівня для слухачів).

Метод складається з однієї чи кількох паралельних активностей;

Активність складається з однієї або кількох послідовних дій, а дія пов'язана з однією або декількома паралельними рольовими частинами, кожна рольова частина асоціює одну з однією діяльністю або діяльністю-структурою. Процес викладання-навчання моделюється у методі з урахуванням поняття театральної п'єси. У п'єсі є акти, й у кожному акті одна чи кілька ролей-частей. Акти в п'єсі

йдуть один за одним у послідовності. Ролі-частини в акті пов'язують кожен роль із діяльністю. Діяльність, своєю чергою, визначає, що ця роль має виконувати і яке середовище доступне нею в рамках дії. За аналогією, призначена діяльність є еквівалентом сценарію для ролі, яку роль грає в акті, хоч і менш приписуючим. Якщо в акті є більше однієї ролі частини, вони виконуються паралельно.

1.2 Система електронного навчання

Система управління навчанням (LMS) – це програмне забезпечення для адміністрування, документації, відстеження, звітності, автоматизації та проведення навчальних курсів, навчальних програм або програм навчання та розвитку. [2] Концепція системи управління навчанням виникла безпосередньо з електронного навчання. Хоча перші LMS з'явилися у 1924 році у секторі вищої освіти, більшість LMS сьогодні зосереджені на корпоративному ринку. Системи управління навчанням складають найбільший сегмент ринку систем навчання. Перше впровадження LMS відбулося наприкінці 1990-х років [3] Системи управління навчанням зазнали значного зростання використання через акцент на дистанційне навчання під час пандемії COVID-19. [4]

Системи управління навчанням були розроблені для виявлення прогалин у навчанні та тренінгах, використовуючи аналітичні дані та звітність. LMS орієнтовані на онлайн-навчання, але підтримують цілий ряд застосувань, виступаючи як платформа для онлайн-контенту, включаючи курси, як асинхронні, так і синхронні. У сфері вищої освіти LMS може пропонувати управління класом для навчання під керівництвом інструктора або "перевернутий клас". Сучасні LMS включають інтелектуальні алгоритми для автоматичного складання рекомендацій з курсів на основі профілю навичок користувача, а також витягують метадані з навчальних матеріалів, щоб зробити такі рекомендації ще точнішими.

LMS може бути розміщена або на локальному хостингу або у постачальника. Хмарна система, розміщена у постачальника, зазвичай працює за моделлю SaaS (програмне забезпечення як послуга). Всі дані в системі,

розміщеній у постачальника, зберігаються у нього, а доступ до них користувачі отримують через Інтернет, з комп'ютера чи мобільного пристрою. Системи, розміщені у постачальника, зазвичай простіше у використанні та вимагають менше технічних знань. У LMS, розміщеній локально, всі дані, що стосуються LMS, зберігаються на внутрішніх серверах користувачів. Програмне забезпечення LMS, розміщене на локальному хостингу, часто має відкритий вихідний код, тобто користувачі отримують (платно чи безкоштовно) програмне забезпечення LMS та його код. Таким чином, користувач може змінювати та підтримувати програмне забезпечення за допомогою внутрішньої команди. Приватні особи та невеликі організації вважають за краще використовувати хмарні системи через вартість внутрішнього хостингу та обслуговування.

За допомогою LMS наставники можуть створювати та інтегрувати матеріали курсу, формулювати цілі навчання, узгоджувати зміст та оцінки, відстежувати прогрес у навчанні та створювати індивідуальні тести для слухачів. LMS дозволяє доносити цілі навчання та організовувати навчальні графіки. Перевага LMS у тому, що вона доставляє навчальний контент та інструменти безпосередньо слухачам, а оцінка може бути автоматизована. Вона також може охоплювати маргіналізовані групи за допомогою спеціальних налаштувань. Такі системи мають вбудовані функції, що настроюються, включаючи оцінку і відстеження. Отже, слухачі можуть у реальному часі свій прогрес, а наставники можуть контролювати й повідомляти про ефективність навчання.[5][6]

Однією з найважливіших особливостей LMS є спроба створити впорядковану комунікацію між слухачами та наставниками. Такі системи, крім полегшення онлайн-навчання, відстеження прогресу у навчанні, надання цифрових інструментів навчання, управління комунікацією та, можливо, продажу контенту, можуть використовуватися для забезпечення різних комунікаційних функцій.[7]

Суміжні поняття включають систему управління контентом (Content Management System), яка відноситься до організації навчального або іншого контенту, а не до загального середовища; систему управління навчальним

контентом (Learning Content Management System), яка найчастіше використовується для корпоративних систем навчання, ніж для систем у навчальних закладах; кероване середовище навчання (MLE), яке зазвичай відноситься до загальної інфраструктури навчального закладу, компонентом якої є VLE; систему підтримки навчання (LSS); центр онлайн-навчання (OLC); чи навчальну платформу (LP); навчання за допомогою комп'ютерно-опосередкованої комунікації (CMC); або онлайн-освіту.

Термін "віртуальне середовище навчання" найчастіше використовується в Європі та Азії, тоді як синонімічний термін "система управління навчанням" є більш поширеним у Північній Америці.

Термін LMS також може означати "система управління бібліотекою" (яка нині найчастіше називається інтегрованою бібліотечною системою, або ILS).

Існує безліч інтеграційних стратегій для вбудовування контенту в LMS, включаючи AICC, xAPI (також званий "Tin Can"), SCORM (Sharable Content Object Reference Model) та LTI (Learning Tools Interoperability)[8]

1.3 Процес електронного навчання на основі стандарту SCORM

Як відомо, у розвитку сучасних технологій важливу роль відіграє стандартизація, завдяки якій стає можливою інтеграція в єдину систему елементів, створених різними виробниками. Зокрема стандарти в галузі інформаційно-комунікаційних технологій в освіті дають змогу електронні освітні ресурси (EOR), розроблені для одних програмних середовищ (платформ), без жодних змін і модифікацій використовувати в інших. Це дає великий ефект, економлячи час і кошти, відкриває можливості для створення репозиторіїв повторно використовуваних EOR [9], [10] на корпоративному, національному та міжнародному рівнях тощо. Останнім часом інформаційні технології в освіті зазнали чималих змін, пов'язаних з масовим використанням мобільних пристроїв: планшетів, смартфонів. Це спонукає до перегляду самих підходів до створення

засобів для е-навчання і робить актуальною проблему розробки і впровадження нових стандартів.

На сьогодні основним стандартом на ринку електронних засобів навчання все ще залишається набір специфікацій під загальною назвою SCORM (Sharable Content Object Reference Model) [11], розроблений ініціативною групою ADL (Advanced Distributed Learning) [12]. Зараз цей стандарт уже явно застарів і не відповідає сучасним потребам, зокрема через те, що передбачає представлення навчального контенту у вигляді архівованих наборів файлів (SCORM-пакетів), які мають знаходитися в тому ж домені, що й система управління навчанням (LMS), яка їх використовує. Очевидно, в епоху Інтернету та хмарних технологій це є анахронізмом. Значно швидше й ефективніше платформи можуть взаємодіяти не через такий обмін файлами, а через взаємне надання послуг веб-сервера, як це реалізовано в розробленому IMS Global Learning Consortium [13] стандарті Learning Tools Interoperability (LTI) [14]. Цей стандарт регламентує прямий обмін даними між віртуальними освітніми середовищами, у процесі якого користувачі одного середовища мають змогу одержувати доступ до ресурсів іншого, що робить можливим реалізацію ідеї об'єднання багатьох середовищ у єдине розподілене освітнє середовище. Приклади практичного використання стандарту LTI розглядалися нами в роботі [15]. Ще більше потенційних можливостей відкриває створений ADL у 2013 р. Новий стандарт – Experience API (скорочено xAPI), раніше відомий також під назвою Tin Can API [16], [17].

Він не просто заміняє SCORM, а й створює передумови для вирішення значно більшої, дійсно глобальної задачі – збору, накопичення в мережевих сховищах Learning Record Store (LRS) і опрацювання даних про результати діяльності особи, дотичних до формування її компетентностей у рамках формальної, неформальної та інформальної освіти впродовж усього життя. При цьому в якості xAPI-клієнтів – постачальників даних для LRS (xAPI-серверів) можуть виступати будь-які програмні засоби: від традиційних LMS – досить складних систем, кожна з яких реалізує повний набір необхідних для організації освітнього процесу функцій, до соціальних мереж і зовсім простих навчальних

програм для мобільних пристроїв. Особливості цього стандарту і приклади його практичного використання досліджувалися нами в роботі [18]. Хоча з часу створення xAPI пройшло шість років, однак очікуваних результатів від його впровадження поки що не спостерігається. На думку більшості дослідників, основна причина цього полягає в тому, що стандарт xAPI є занадто гнучким, що ускладнює його практичне застосування. З метою подолання цього недоліку в 2016 р. ADL запропонував певне уточнення цього стандарту у вигляді профілю (окремого випадку) специфікації xAPI, який одержав назву smi5. Зараз на ринку вже почали з'являтися перші програмні засоби з підтримкою smi5, однак у вітчизняній науковій літературі вони поки що не досліджувалися.

Sharable Content Object Reference Model (SCORM) – стандарт, розроблений для систем дистанційного навчання. Цей стандарт містить вимоги до організації навчального матеріалу та всієї системи дистанційного навчання. SCORM дозволяє забезпечити сумісність компонентів та можливість їх багаторазового використання: навчальний матеріал представлений окремими невеликими блоками, які можуть включатись у різні навчальні курси та використовуватися системою дистанційного навчання незалежно від того, ким, де та за допомогою яких засобів були створені. SCORM базується на стандарті XML.

Відповідно до цих стандартів послідовності та навігації, правила послідовності навчання та стратегії презентації для електронного навчання кодуються у XML для вбудовування в елементи триструктурних організацій контенту. Тим не менш, деталі специфікацій SN на основі XML надзвичайно складні, і розробникам змісту дуже важко зрозуміти та скласти свої пакети змісту електронного навчання. Попередня версія SCORM також не визначає жодної інформації про послідовність, щоб учень міг вибирати, який навчальний об'єкт використовувати і в якому порядку. Тому, безумовно, необхідно підкріпити його певним чином зручними засобами щодо послідовності дій електронного навчання та навігаційних специфікацій на даний момент.

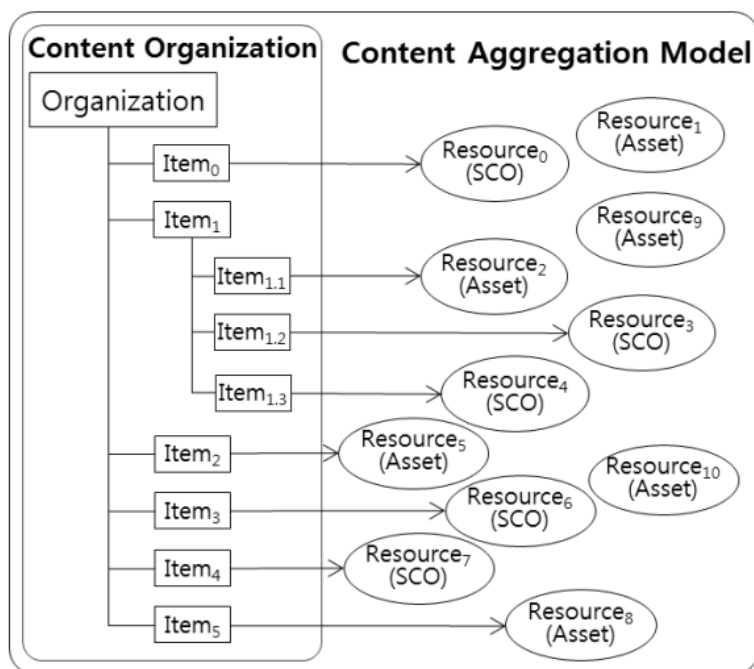


Рисунок 1.2 – Структура моделі агрегації контенту

Як показано на рисунку 1.2, основна структура моделі агрегації контенту [19][20] складається з організацій контенту та об'єктів навчання. Організація контенту будується деревоподібною структурою елементів, а навчальні об'єкти представлені ресурсами, кожен із яких пов'язані з SCO (Sharable Content Object) чи активом, вибраним з бункера навчальних об'єктів. Більш того, ці елементи моделі агрегації контенту точно визначені пакетом SCORM. Тобто специфікація упаковки LMS (e-Learning Management System) використовує SCORM як основу для пакування та організації навчальних об'єктів. Пакет може містити більше однієї організації контенту для тих самих об'єктів навчання, що передбачає, що кілька процесів електронного навчання, що охоплюють один і той же предмет на різних рівнях глибини або для різних аудиторій, можуть бути визначені в одному пакеті SCORM. Таким чином, LMS може скористатися гнучкістю цієї функції, що дозволяє слухачам вибирати більш відповідні організації контенту електронного навчання.

Модель управління процесом e-Learning заснована на математичному формалізмі, що дозволяє графічне проектування процесу навчання, керованого попередніми вимогами, для специфікацій послідовності SCORM, і вона націлена

на те, щоб зрештою стати теоретичною основою для розробки системи моделювання послідовності SCORM, заснованої на попередніх вимогах. Система моделювання послідовності або графічна система моделювання агрегації контенту електронного навчання, здатна автоматично генерувати специфікації контенту електронного навчання на основі SCORM для системи управління електронним навчанням на основі SCORM. У цьому розділі описується фундаментальна мотивація моделі та представлена метамодель та опис її компонентів.

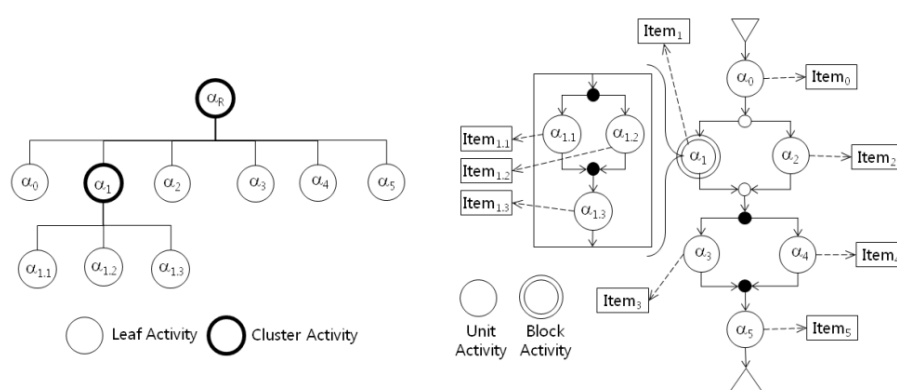


Рисунок 1.3 – Дерево діяльності електронного навчання проти моделі керування процесом електронного навчання

У поточному стандарті SCORM організація контенту є картою, яка відображає передбачуване використання контенту через деревоподібні одиниці інструкцій (Activities). Карта показує, як події пов'язані один з одним, як показано в лівій частині рисунку 1.3

Кожен із видів діяльності відповідає елементу організації контенту в моделі агрегації контенту, показаної на рисунку 1.2, і концептуально можливо, щоб певний вид діяльності містив інші види діяльності, що називається кластерною діяльністю у дереві діяльності SCORM. Також немає обмежень на кількість рівнів вкладеності для видів діяльності. Лише листова діяльність має пов'язаний із нею навчальний ресурс (SCO чи Asset), який використовується для виконання відповідного елемента. У термінах SCORM визначено розширений елемент Prerequisite маніфесту елемента та модель послідовності та навігації (SN).

Інформація про послідовність, така як передумови або елементи послідовності моделі SN, яку називають процесом електронного навчання, вбудовані у маніфест кожного елемента як внутрішні специфікації кожної діяльності.

Коротко кажучи, модель агрегації контенту SCORM надає спосіб представлення процесу електронного навчання через дерево діяльності або елемент передумови, а з іншого боку, може бути надано певний тип автоматизованих засобів для визначення та візуалізації набору процесів електронного навчання, що протікають над організацією контенту електронного навчання. Без таких автоматизованих засобів дуже важко і незручно визначати та візуалізувати процеси електронного навчання, а також використовувати деякі просунуті функції послідовності, такі як динамічні зміни, альтернативна та паралельна послідовність, циклічна послідовність, звітність та аудит стану процесів електронного навчання тощо.

1.4 Модель керування процесом електронного навчання SCORM

Процес електронного навчання – це заздалегідь визначений чи запланований набір кроків електронного навчання, які називаються діями, та частковий порядок цих дій. Система управління електронним навчанням допомагає організувати, контролювати та виконувати екземпляри електронного навчання, які називаються e-Learning-cases, створені на основі процесу електронного навчання. Аспект управління процесом електронного навчання задається упорядкованими комбінаціями послідовних (перед тим, як приступити до діяльності А, завершіть діяльність В), кон'юнктивних (перед тим, як приступити до діяльності А, завершіть діяльність В та С), диз'юнктивних (перед тим, як приступити до діяльності А); діяльності А, завершіть діяльність В чи С) логік передумов. Кожен член передумови визначається статусом передумов, такими як пройдено, завершено, переглянуто, провалено, не пройдено, не пройдено і незавершено. Діяльність електронного навчання є базовою одиницею навчальної поведінки і складається з двох типів - елементарної діяльності та

блокової діяльності, що містить інший процес електронного навчання. Тобто, елементарна активність та блокова активність реалізуються одиничним елементом та складовим елементом, відповідно. Тільки одиничний елемент (елементарна діяльність) матеріалізується через ресурс електронного навчання (SCO або Asset), який використовується для виконання своєї діяльності в одному з трьох режимів: ручний, автоматичний або гібридний. Крім попередньої умови, кожен вид діяльності може додатково визначати нахненні AICC/СМІ розширення для поведінки, пов'язаної з послідовністю, такі як елемент, пов'язаний з часом (максимально допустимий час, обмеження часу дії), елемент даних з eLMS і елемент оцінки майстерності. Елемент, пов'язаний з часом, дозволяє встановити максимальний час для виконання завдання та визначає дію, яка відбудеться, якщо учень не вкладеться у відведений час; дані з елемента eLMS дозволяють визначити, як буде надіслано повідомлення. Елемент mastery score використовується для визначення прохідного балу, який повинен мати значення від 0 до 100 для даного виду діяльності. На рисунку 1.4 відображено мета-модель управління процесом електронного навчання.

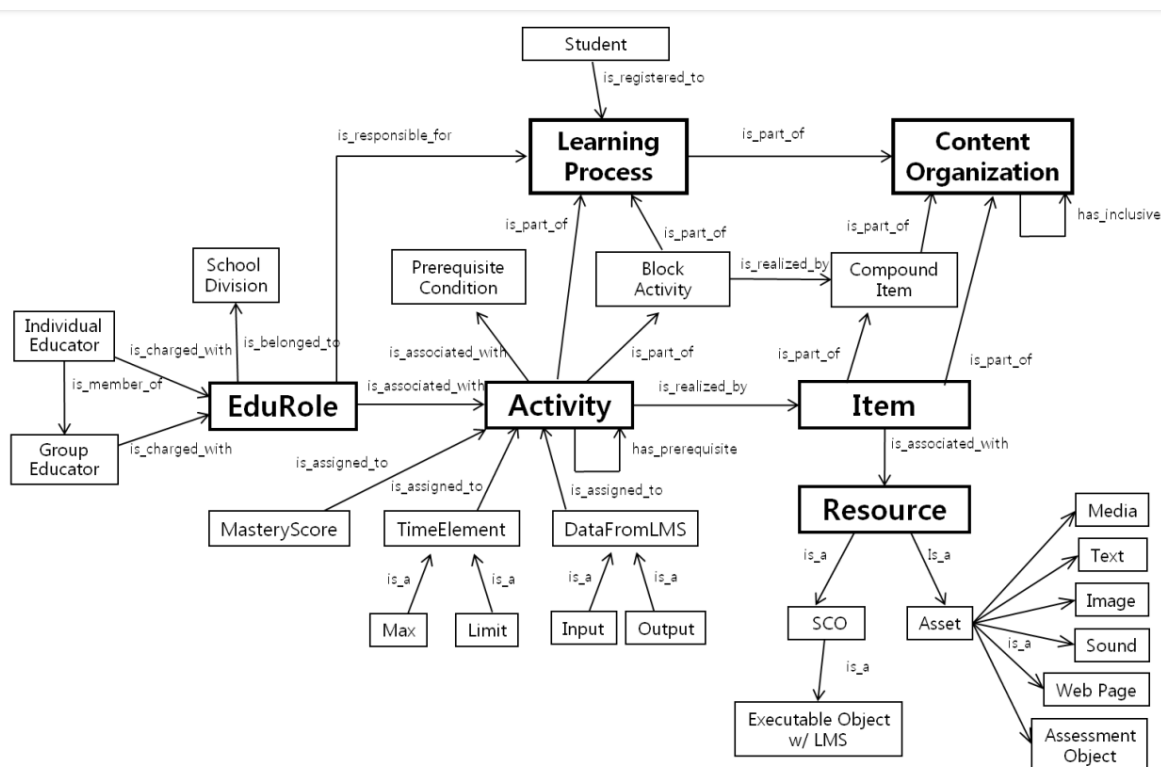


Рисунок 1.4 – Мета-модель управління процесом електронного навчання

Як правило, один або кілька учасників (менторів) пов'язані з кожним видом діяльності у вигляді ролей. Роль EduRole - це іменоване позначення для одного або кількох педагогів, яке зручно виступає як основна відповідальність відповідної діяльності. Педагог - це людина чи група, які можуть виконувати відповідну роль до виконання, відповідальності чи якимось чином асоціюватися з діяльністю та процесами електронного навчання. Декілька копій, так званих e-Learning-кейсів, одного і того ж процесу електронного навчання можуть перебувати на різних стадіях виконання. Отже, процес електронного навчання можна як клас, а кожен e-Learning-case - як екземпляр класу, кожен із яких проходить слухач. Таким чином, e-Learning-case окреслюється місцем контролю для конкретного виконання процесу електронного навчання. Якщо навчання електронного навчання виконується в автоматичному або гібридному режимі, це означає, що весь/частина ресурсу SCO (або Asset), пов'язаного з цим заняттям, автоматично запускається службою e-Learning enactment service.

1.5 Математична модель управління процесом електронного навчання

На основі мета-моделі, описаної на рисунку 1.4, можемо визначити її формальне уявлення моделі управління процесом електронного навчання. Модель може бути використана для автоматичного генерування моделі агрегації контенту SCORM, що включає концепцію процесу електронного навчання з передумовами послідовності на прикладі створення типового курсу електронного навчання.

Базова модель управління процесом електронного навчання формально визначається 9-кортежним $\Gamma = (\delta, \varphi, \xi, \gamma, \varepsilon, \pi, \kappa, I, O)$, [21] що формується над безліччю видів діяльності електронного навчання A (включаючи діяльність за блоками/одинацями), безліччю предметів S , множиною R ресурсів, реалізованих SCO, множиною C необхідних умов. елементів, безлічі R ресурсів, що реалізуються SCO або асетами(Assets), безлічі C необхідних умов, набір D

релевантних даних з eLMS, набір Р освітніх ролей і набір Т наставників. (включаючи індивідуальних та групових наставників), де

– I це кінцевий набір вихідних релевантних даних, які, як передбачається, завантажуються інформацією зовнішнім процесом електронного навчання перед виконанням моделі;

– кінцева множина остаточних вихідних релевантних даних, що містять інформацію, яка використовуватиметься деяким зовнішнім процесом електронного навчання після виконання моделі;

– $\delta = \delta_i \cup \delta_o$ де, $\delta_o : \mathbf{A} \rightarrow \wp(\mathbf{A})$ це багатозначна функція відображення електронного навчального заняття ($\alpha \in \mathbf{A}$) на безліч його безпосередніх наступників з постпререквізитами в поведінці послідовності електронного навчання, а $\delta_i : \mathbf{A} \rightarrow (\mathbf{A})$ - це багатозначна функція відображення діяльності на її безліч безпосередніх попередників з пререквізитами в поведінці послідовності електронного навчання;

– $\varphi = \varphi_{item} \cup \varphi_a$ де, $\varphi_{item} : \mathbf{A} \rightarrow \wp(\mathbf{S})$ однозначна функція відображення електронного навчального завдання ($\alpha \in \mathbf{A}$) на набір елементів електронного навчання, визначений у моделі організації контенту, та $\varphi_a : \mathbf{S} \rightarrow \wp(\mathbf{A})$ це багатозначна функція відображення елемента електронного навчання на множину видів діяльності електронного навчання;

– $\xi = \xi_{resource} \cup \xi_{item}$ де, $\xi_{item} : \mathbf{R} \rightarrow \wp(\mathbf{S})$ це однозначна функція відображення ресурсу ($\sigma \in \mathbf{R}$) до набору елементів електронного навчання, визначеного в моделі організації контенту, та $\xi_{resource} : \mathbf{S} \rightarrow \wp(\mathbf{R})$ це однозначна функція відображення елемента електронного навчання ($\beta \in \mathbf{S}$) на набір потужностей ресурсів;

– $\gamma = \gamma_i \cup \gamma_o$ де, $\gamma_o : \mathbf{A} \rightarrow \wp(\mathbf{D})$ це багатозначна функція відображення електронного навчального завдання на його набір вихідних релевантних даних, та $\gamma_i : \mathbf{A} \rightarrow \wp(\mathbf{D})$ це багатозначна функція відображення електронного навчального завдання на його набір релевантних вхідних даних; Де тип релевантних даних

класифікується на оцінку майстерності, максимально допустимий час, обмеження часу дії та дані LMS;

– $\varepsilon = \varepsilon_a \cup \varepsilon_p$ де, $\varepsilon_p : \mathbf{A} \rightarrow \mathbf{P}$ це однозначна функція відображення електронного навчального завдання на одну з освітніх ролей, and $\varepsilon_a : \mathbf{P} \rightarrow \wp(\mathbf{A})$ це багатозначна функція відображення освітньої ролі на набір можливостей електронного навчання;

– $\pi = \pi_p \cup \pi_t$ де, $\pi_t : \mathbf{P} \rightarrow \wp(\mathbf{T})$ є багатозначною функцією відображення ролі педагога на безліч педагогів, та $\pi_p : \mathbf{T} \rightarrow \wp(\mathbf{R})$ це багатозначне відображення педагога на набір освітніх ролей;

– $\kappa = \kappa_i \cup \kappa_o$ де, $\kappa_i : \mathbf{A} \rightarrow \wp(\mathbf{C})$ це безліч передумов, \mathbf{C} , на кожній дузі, $(\delta_i(\alpha), \alpha)$, $\alpha \in \mathbf{A}$; та $\kappa_o : \mathbf{A} \rightarrow \wp(\mathbf{C})$ - це безліч постреквізитних умов, \mathbf{C} на кожній дузі, $(\alpha, \delta_o(\alpha))$, $\alpha \in \mathbf{A}$; Де значення кожного елемента безлічі попередніх/постреквізитних умов є одним із значень статусу: {пройдено, завершено, переглянуто, провалено, не пройдено, не намагалися і не закінчено}.

Запуск та завершення діяльності електронного навчання. Крім того, виконання моделі управління процесом електронного навчання починається з однієї умови переходу λ Таким чином, завжди припускаємо без втрати спільності, що існує єдина початкова дія електронного навчання (α_I) . На початку передбачається, що всі релевантні вхідні дані в наборі, \mathbf{I} , були ініціалізовані даними зовнішньої системи:

$$\{ \exists \alpha_I \in A \mid \delta_i(\alpha_I) = \emptyset \wedge \kappa_o(\alpha_I) = \{\lambda\} \} \quad (1.1)$$

Виконання завершується за будь-якої умови переходу λ на виході. Існує єдина завершальна діяльність електронного навчання (α_F) . Набір вихідних сховищ, \mathbf{O} , є власниками даних, які можуть бути використані після завершення роботи зовнішньою системою:

$$\{ \exists \alpha_F \in A \mid \delta_o(\alpha_F) = \emptyset \wedge \kappa_i(\alpha_F) = \{\lambda\} \}. \quad (1.2)$$

Послідовність поведінки із структурованими властивостями. Враховуючи формальне визначення, поведінку послідовності в моделі управління процесом електронного навчання можна інтерпретувати в такий спосіб.

Для будь-якої діяльності електронного навчання $\alpha \in A$, у загальному випадку,

$$\delta_i(\alpha) = \{ \begin{array}{l} \{ \beta_{11}, \beta_{12}, \dots, \beta_{1m(1)} \}^s, \\ \{ \beta_{21}, \beta_{22}, \dots, \beta_{2m(2)} \}^s, \\ \dots \\ \{ \beta_{n1}, \beta_{n2}, \dots, \beta_{nm(n)} \}^s \\ \end{array} \}$$

означає, що перед тим, як розпочати діяльність з електронного навчання (α), він повинен бути попередньо задоволений однією з таких ситуацій. Де s - селективний номер, значення якого може бути 0, 1 або x ($1 < x < n$).

– $n = 1 \wedge m(n) = 1 \wedge s = 0$: Послідовний пост-реквізит, який пов'язаний з послідовною поведінкою. Наприклад, $\delta_i(\alpha) = \{\{\beta\}^0\}$ представляє, що β пов'язаний з передумовою α . Тобто, після завершення α , діяльність з електронного навчання, β , здатна пообідати власні предмети електронного навчання.

– $n > 1 \wedge \forall m(i) = 1 (1 \leq i \leq n) \wedge s = 1$: Кон'юнктивний (I) пост-реквізит, який пов'язаний з паралельною поведінкою (паралельний-спліт). Як приклад, $\delta_i(\alpha) = \{\{\beta_{1j}\}^1, \{\beta_{2j}\}^1, \dots, \{\beta_{nj}\}^1, 1 \leq j \leq (m(j))\}$ представляє, що всі дії з електронного навчання, $\{\beta_{1j}\}^1, \{\beta_{2j}\}^1, \dots, \{\beta_{nj}\}^1, 1 \leq j \leq (m(j))$, повинні бути одночасно розпочаті після завершення діяльності з електронного навчання, α . [21]

– $n = 1 \wedge m(n) > 1 \wedge s = 1$: Диз'юнктивний (Exclusive-OR) пост-реквізит, який пов'язаний з альтернативною поведінкою (альтернативний-спліт). Наприклад, $\delta_i(\alpha) = \{\{\beta_{i1}, \beta_{i2}, \dots, \beta_{im(i)}\}^1, 1 \leq i \leq n\}$ представляє, що лише один з заходів електронного навчання, $\beta_{i1}, \beta_{i2}, \dots, \beta_{im(i)}, 1 \leq i \leq n$, повинен бути обраний і запущений після завершення діяльності електронного навчання, α . [21]

– $n = 1 \wedge m(n) > 1 \wedge s = x$: Селективне з'єднання (X^* : Селективне-I) пост-

реквізит, що має на увазі селективну паралельну поведінку (селективний-спліт). Наприклад, $\delta_i(\alpha) = \{\{\beta_{i1}, \beta_{i2}, \dots, \beta_{im(i)}\}^x\}$ з встановленням селективного числа представляє, що кілька вибірових варіантів, x ($1 < x < n$), з діяльності електронного навчання, $\beta_{i1}, \beta_{i2}, \dots, \beta_{im(i)}$, $1 \leq i \leq n$, повинні бути одночасно розпочаті відразу після завершення діяльності електронного навчання, α . [21]

Зокрема, при формуванні структурованих типів (пара split та join) поведінок секвенування, таких як AND, Exclusive-OR та Selective-AND, у моделі управління процесом електронного навчання, вони повинні бути правильно структуровані шляхом збереження структурних властивостей, властивості правильного вкладення та властивості підібраних пар.

1.6 Огляд існуючих систем та програмних рішень

Тенденція переходу від традиційних аудиторних курсів до онлайн-тренінгів слідує за переходом теорій навчання від поведінкової орієнтації, яка зображує навчання як переважно пасивну діяльність, до теоретичної орієнтації, яка наголошує на активній, рефлексивній та соціальній природі навчання. У міру розвитку цієї тенденції виникають питання щодо вибору системи управління навчанням (LMS) для розробки онлайн-курсу. Вибір LMS має вирішальне значення для успіху слухачів. Цей вибір має бути заснований як на цілях онлайн-курсу, так і на потребах слухачів. LMS повинна мати компоненти, які дозволять наставнику створити курс із акцентом на активне навчання та засвоєння матеріалу.

Кожна програма LMS ґрунтується на здатності пристосовувати різні активні види навчання в онлайн-курсах. Системи LMS для порівняння: Moodle v3.10; WebCT; BlackBoard, v. 9.1; Educator; ISpring; та Sakai v20.1. Система управління навчанням має задовольняти потреби кінцевого користувача - слухача. Кожна LMS має ключові особливості, які дозволяють слухачам брати активну участь у своїх курсах. Ключові функції різних систем управління навчанням розширюють можливості викладання та навчання та залучають слухачів до глибшого

навчального процесу. Під час вибору та використання будь-якого типу навчального програмного забезпечення наставнику необхідно мати компоненти, які дозволять йому створювати курси активного навчання.

Завдяки зручності використання LMS стає завданням номер один як для наставників, так і для слухачів. Відповідно до [23], зручність використання – це "ступінь, в якому продукт може бути використаний конкретними користувачами для досягнення певних цілей з ефективністю, результативністю та задоволенням у певному контексті використання". Зручність використання – ключовий момент в онлайн-курсі. Ефективність курсу допоможе слухачам досягти конкретних цілей курсу. Простота навігації за курсом допоможе слухачам досягти своєї мети. Якщо курс не є ефективним чи неефективним, це вплине на навчання слухачів. Кожен компонент LMS є важливим для навчання слухача. Правильно підібрані компоненти можуть зробити курс кращим, а неправильні – зруйнувати його. Нижче наводиться порівняння кожної LMS за компонентами системи

Розробка контенту відіграє важливу роль у розробці курсу. Ключовим моментом розробки контенту є ефективність і результативність LMS. Курс має бути орієнтований на слухача. Підхід, орієнтований на слухачів, підтримує слухачів та мотивує їх до навчання. Що помістити в курс або як завантажити контент має вирішальне значення. Наставники повинні мати можливість завантажувати в онлайн-курс не лише конспекти лекцій. Вони повинні мати можливість завантажувати та редагувати контент у режимі онлайн. Вони також повинні створювати контент з урахуванням здатності слухачів обробляти та розуміти зміст курсу. Розробка контенту повинна бути послідовною і логічно збудованою таким чином, щоб слухачі могли легко дотримуватися курсу. Тому наставнику слід обрати LMS, яка підтримує підхід, орієнтований на слухача. Вона має дозволяти наставнику розробляти високоякісний матеріал. LMS повинна включати механізми підтримки, доступні для тих, хто навчається в режимі онлайн. На таблиці 1.1 відображено порівняльну характеристику існуючих середовищ за можливістю розробки навчального контенту.

Таблиця 1.1 – Розробка контенту курсу

Можливості для інструкторів	Moodle	WebCT	Blackboard	Educator	ISpring	Sakai
Вибір різних компонентів курсу	✓	✓	✓	✓	✓	✓
Створення копії існуючого курсу	✓	✓	✓	✓	✓	
Приховання поточної роботи	✓	✓	✓	✓	✓	
Завантаження файлів з робочого столу на сервер	✓	✓	✓	✓	✓	✓
Завантаження кількох файлів одночасно у вигляді архіву		✓	✓	✓	✓	
Редагування завантаженого тексту онлайн		✓	✓		✓	✓
Реліз контенту на основі перегляду слухачами певних попередніх матеріалів		✓	✓		✓	
Редагування сторінок контенту на підставі результатів виконання тестів слухачів		✓		✓		✓
Забезпечення зворотного зв'язку в рамках тестів		✓			✓	✓
Випуск вмісту для певних груп	✓	✓	✓	✓	✓	✓
Створення посилань	✓	✓	✓		✓	✓
Використання автоматизованого інструменту для створення глосарію	✓	✓	✓	✓	✓	
Використання автоматизованого інструменту для створення бази даних зображень з можливістю пошуку	✓	✓	✓	✓	✓	✓
Прикріплення цілей навчання до сторінок контенту	✓	✓	✓	✓	✓	✓

Системи LMS мають вбудований календар подій, котрий відображає всі минулі, теперішні та майбутні події котрі створює ментор. Навчаємі, котрим треба пройти тренінг, автоматично додаються та отримують посилання на цей тренінг з календарю. На таблиці 1.2 відображено порівняльну характеристику існуючих середовищ за взаємодією з календарем тренінгу. Наявність навчальних посібників допоможе слухачам вивчити матеріали курсу.

Таблиця 1.2– Взаємодія з календарем курсу

Можливість взаємодії з календарем	Moodle	WebCT	Blackboard	Educator	ISpring	Sakai
Календар на сторінці входу відображає події з усіх курсів	✓	✓	✓	✓	✓	✓
Оголошення зі всіх курсів відображаються на сторінці входу до системи	✓		✓		✓	
Інструктор може завантажити кілька подій із файлу		✓			✓	
Інструктор може додавати посилання до календарних записів.	✓	✓	✓		✓	✓
Слухачі можуть додавати записи до календаря курсу календар (на розсуд наставника)	✓	✓		✓	✓	

Крім того, якщо інструктор створює для слухачів тест для самоперевірки, вони можуть використовувати його як навчальний посібник з курсу. Однією з основних функцій, яку слухачі цінують у LMS, є система, що дозволяє їм шукати певний контент. Наприклад, якщо слухачі вивчає квадратні рівняння, він може знайти у змісті курсу всі сторінки, присвячені квадратним рівнянням. На таблиці 1.3 відображено порівняльну характеристику існуючих середовищ за доступними інструментами для користувачів тренінгу.

Таблиця 1.3 – Інструменти для навчаємих

Можливості слухача	Moodle	WebCT	Blackboard	Educator	ISpring	Sakai
Створення навчального посібника		✓				✓
Продовжити читання вмісту з останнього місця зупинки		✓			✓	✓
Додати нотатки до змісту курсу Веб-сторінки		✓			✓	✓
Робити нотатки в інтегрованому блокноті		✓	✓	✓	✓	✓
Скласти набір повідомлень електронної пошти		✓		✓		
Скласти набір оголошень на дошці оголошень	✓	✓	✓	✓		
Пройти тести для самоперевірки, створені викладачем	✓	✓		✓	✓	✓
Пошук матеріалів курсу		✓	✓		✓	✓

У традиційному курсі наставник може враховувати відвідуваність тренінгів. Проте він не знає, чи слухачі читають чи звертають увагу на дискусії. В онлайн-тренінгу ментори можуть уважно стежити за слухачами, таблиця 1.4. Вони можуть визначити, коли слухач заходить у курс і як довго він перебуває в курсі у певний час. Вони також можуть відстежувати, скільки обговорень слухач читає та на що відповідає. Вони також можуть відстежувати, коли та скільки сторінок контенту переглядає слухач. Наставники можуть оцінювати роботу слухачів через області надсилання та зараховувати прогрес. Після того, як ментор оцінив проходження тренінгу, слухачі можуть переглянути свої результати та будь-які відгуки про завдання.

Таблиця 1.4 – Контроль участі та прогресу навчаємих

Можливості	Moodle	WebCT	Blackboard	Educator	ISpring	Sakai
Ментори можуть аналізувати клас за комбінаціями інформації, наприклад, слухачі, які увійшли до системи після певного дня та розмістили певну кількість статей	✓	✓	✓	✓	✓	✓
Ментори можуть переглянути, скільки статей на дошці оголошень прочитав слухач		✓			✓	
Ментори можуть переглянути, скільки статей на дошці оголошень опублікував слухач		✓	✓	✓	✓	✓
Слухач може переглядати власні оцінки за тренінги	✓	✓	✓	✓	✓	✓
Наставники можуть відстежити кількість переглядів на сторінку	✓	✓	✓	✓	✓	✓
Наставники можуть переглядати дату та час кожного першого та останнього вхід у систему кожного слухача		✓				
Наставники можуть переглядати повну історію час, проведений кожним слухачем в онлайні протягом усього курсу	✓		✓		✓	✓

Система управління навчанням має задовольняти потреби кінцевого користувача - навчаємого. Кожна LMS має ключові особливості, які дозволяють слухачам брати активну участь у своїх курсах. Різні системи управління

навчанням розширюють можливості викладання та навчання та залучають учнів до глибшого навчального процесу. Зручність використання системи управління навчанням – це ключ до ефективності та результативності онлайн-курсу. Викладачі повинні розуміти різні компоненти LMS і те, як ці компоненти працюватимуть, щоб зробити курс ефективним та результативним. LMS повинна бути здатна легко та ефективно створювати педагогічно обґрунтований контент курсу та навчальні об'єкти. Вона повинна мати всі необхідні інструменти електронного навчання для оцінки, комунікації, співробітництва та створення спільноти, а також для створення та управління онлайн-навчальними курсами.

1.7 Висновки до першого розділу

В першому розділі проведено аналіз предметної області, а саме електронного навчання. Визначено головні принципи електронного навчання та переваги перед традиційним навчанням.

Проаналізовано принципи проектування та основні можливості систем управління навчанням. Системи управління навчальним процесом були розроблені для виявлення прогалин у дистанційному навчанні на базі використання аналітичних та фактичних даних. LMS призначені для онлайн-навчання, але підтримують цілий ряд додатків, виступаючи як платформа для онлайн-контенту, включаючи як асинхронні, так і синхронні курси та тренінги.

Проаналізовано переваги та функціональність існуючих LMS, а саме таких як: Moodle v3.10; WebCT; BlackBoard, v. 9.1; Educator; ISpring; та Sakai v20.1.

2 ДОСЛІДЖЕННЯ МОДЕЛІ ТА МЕТОДУ ТЕСТУВАННЯ

2.1 Постановка задачі

Основною проблемою в електронному навчанні є оцінювання як інструмент вимірювання знань слухачів. В даний час не існує широко використовуваних інтегрованих і комплексних рішень, що поєднують весь процес створення, опису, зв'язування та надання адаптивних онлайн-тренінгів та тестів, що є основною мотивацією виконання даної дослідницької роботи.

У зв'язку з цим пропонується інтелектуальна система генерації та оцінювання електронних навчальних тестів з використанням цілочисельного програмування, запропоновано концептуальну мета-модель оцінювального простору, що включає інструменти для створення інтерактивних ресурсів та оцінки накопичених попередніх знань.

Описана система відрізняється від інших способів створення тестів тим що, спрямована на визначення ряду питань різного ступеню складності із попередньо визначеного набору питань, які складатимуть тест, дозволяє генерувати і оцінювати тести різного рівня складності. Модель призначена для створення та надання персоналізованого електронного навчання на основі інтеграції аналітичного комплексу, що включає інструменти оцінки, базу даних необхідних компетенцій, зміст навчання, а також попередні знання та навички слухачів. Таким чином, слухачеві надається ефективний спосіб пройти весь процес навчання та придбати бажаний профіль компетенцій.

Для реалізації відбору питань для різних рівнів тестів запропоновано дві оптимізаційні моделі, особливостями котрих є використання лінійного цілочисельного програмування.

Перша з них визначає мінімальну кількість запитань шляхом вибору серед складніших питань, а друга має на меті максимізувати кількість запитань шляхом вибору серед менш складних питань. Запропонувати прототип інтелектуальної

системи для генерування та оцінювання електронних навчальних тестів для тренінгів різного рівня складності, котрий реалізовано у вигляді веб-додатка.

З огляду на це метою даної роботи є пропозиція створення прототипу інтелектуальної системи для створення та оцінки тестів електронного навчання з використанням цілісного програмування. Основою цієї системи є формулювання математичних моделей для генерації тестів з більшою кількістю питань, але з меншим ступенем складності, і для генерації тестів з меншою кількістю питань, але з більшим ступенем складності.

Для реалізації цього, здавалося б, не дуже складного процесу необхідна послідовність низки дій щодо фрагментації та зв'язування змісту тренінгу, які включають такі шість етапів:

– Етап 1. Навчальні об'єкти декомпонуються на невеликі одиниці навчального контенту.

– Етап 2. Навчальні об'єкти описуються з допомогою необхідних компетенцій.

– Етап 3. Кожен об'єкт навчання пов'язані з іншими об'єктами навчання.

– Етап 4. Побудова оцінних/аналітичних компонентів.

– Етап 5. Розробка профілю компетенції слухача.

– Етап 6. Індивідуальним слухачам надається доступ до навчального контенту та тестів на основі профілю компетенції.

Майбутня робота буде зосереджена на включенні запропонованої моделі в інтерактивну програмну систему, здатну створювати та надавати зміст електронного навчання відповідно до конкретних потреб та попередніх знань слухачів. У цьому напрямку важливою областю, що визначає нові можливості та завдання для майбутніх досліджень, є розробка відповідного середовища електронного навчання для слухачів із застосуванням запропонованої моделі.

Зрештою, слід зазначити, що перехід до персоналізованого навчання на основі попередніх знань слухачів – це процес, який, безумовно, стане майбутнім електронного навчання. В даний час через комплекс технологічних та нетехнологічних факторів цей процес складний і неефективний. З розвитком

ефективних рішень у цій галузі, що підтримуються передовими технологіями, такими як штучний інтелект та аналіз великих даних, нові досягнення поступово усуватимуть технологічні проблеми і як їх наслідок - знижувати витрати та підвищувати мотивацію як розробників контенту персоналізованого навчання, так і самих слухачів.

Задля виконання поставленої мети потрібно виконати ряд задач, які перераховані нижче:

- Проведення аналізу існуючих видів тестування знань навчаємих і рішень створених на їх основі, виявлення їх переваг і недоліків.

- Створення математичних моделей та проведення теоретичних досліджень на їх основі.

- Проектування та розробка прототипу веб-додатка.

- Дослідження практичного моделювання запропонованих моделей та методів.

2.2 Аналіз переваг та недоліків методів тестування знань слухачів

У порівнянні з іншими формами контролю знань тестування має свої переваги та недоліки.

Переваги:

- Тестування є більш якісним та об'єктивним способом оцінювання, його об'єктивність досягається шляхом стандартизації процедури проведення, перевірки показників якості завдань та тестів.

- Тестування — більш справедливий метод, воно ставить всіх навчаємих у рівні умови, як у процесі контролю, і у процесі оцінки, практично виключаючи суб'єктивізм викладача. За даними англійської асоціації NEAB, що займається підсумковою атестацією навчаємих Великобританії, тестування дозволяє знизити кількість апеляцій більш ніж утричі, зробити процедуру оцінювання однаковою

для всіх учнів незалежно від місця проживання, типу та виду освітньої установи, де займаються учні.

– Тести це більш об'ємний інструмент, оскільки тестування може включати завдання по всіх темах тренінгу, в той час як на усний іспит зазвичай виноситься 2-4 теми, а на письмовий - 3-5. Це дозволяє виявити знання навчаємого по всьому тренінгу, виключивши елемент випадковості під час витягування білету. З допомогою тестування можна встановити рівень знань навчаємого з предмету в цілому та з окремими його темами.

– Тест це точніший інструмент, так, наприклад, шкала оцінювання тесту з 20 питань, складається з 20 поділів.

– Тестування ефективніше з економічної точки зору. Основні витрати під час тестування припадають розробку якісного інструментарію, тобто мають разовий характер. Витрати на проведення тесту значно нижчі, ніж при письмовому або усному контролі. Проведення тестування та контроль результатів у групі з 30 осіб займає півтори дві години, усний чи письмовий іспит – не менше чотирьох годин.

– Тестування – це м'якіший інструмент, вони ставлять усіх слухачів у рівні умови, використовуючи єдину процедуру та єдині критерії оцінки, що призводить до зниження передекзаменаційних нервових напруг.

Недоліки:

– Розробка якісного тестового інструментарію – тривалий, трудомісткий та дорогий процес.

– Дані, отримані викладачем у результаті тестування, хоч і включають інформацію про прогалини у знаннях по конкретних розділах, але не дозволяють судити про причини цих прогалин.

– Тест не дозволяє перевіряти та оцінювати високі, продуктивні рівні знань, пов'язані з творчістю, тобто імовірнісні, абстрактні та методологічні знання.

– Широта охоплення тем у тестуванні має і зворотний бік. Учень при тестуванні, на відміну усного чи письмового іспиту, немає достатньо часу для глибокого аналізу теми.

– Забезпечення об'єктивності та справедливості тесту потребує вживання спеціальних заходів щодо забезпечення конфіденційності тестових завдань. При повторному застосуванні тесту бажано внесення до завдань змін.

– У тестуванні є елемент випадковості. Наприклад, учень, який не відповідає на просте запитання, може дати правильну відповідь на складнішу. Причиною цього може бути як випадкова помилка в першому питанні, так і вгадування відповіді в другому. Це спотворює результати тесту і призводить до необхідності обліку складової ймовірнісної під час їх аналізу

Для вирішення більшості негативних факторів пов'язаних із тривалістю створення тесту та його оцінювання, запропоновано мета-модель оцінювального простору для тренінгу та алгоритм генерації та оцінювання тестів.

2.3 Мета-модель оцінювального простору для тренінгу

Наукова новизна дослідження полягає в тому, що в роботі вдосконалено мета-модель та метод оцінювання для професійного тренінгу, рисунок 2.1.

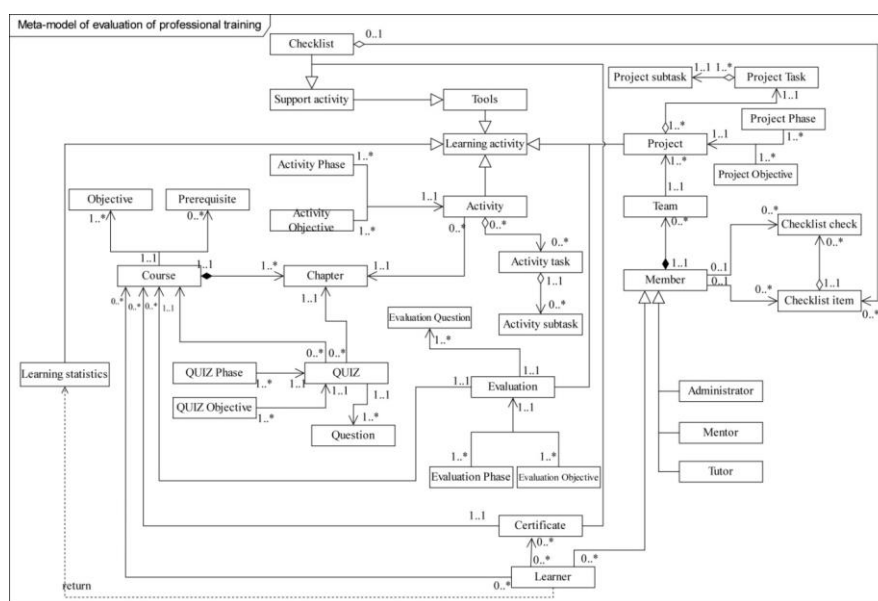


Рисунок 2.1 – Мета модель оцінювального простору для професійного тренінгу

Модифікована модель базується на основній концепції специфікації дизайну навчання, описаній раніше в пункті 1.1. Поточна специфікація навчального оцінювання відображає модель одного користувача, слухача, який взаємодіє з контентом і проходить тестування.

Модель відрізняється тим що забезпечує можливість проектування одиниць навчання, що одночасно включають кілька ролей, кожна з яких може бути зіграна кількома учасниками. Вона дозволяє визначити їхню діяльність у скоординованих "потоках навчання", які аналогічні робочим процесам групового ПЗ. Таким чином, вона підтримує групове і спільне навчання різних видів, важливість якого все більше визнається як в комерційному навчанні, так і в освітній сфері. Вона все ще може бути використана для підтримки моделі з одним навчаємим шляхом створення одиниці навчання з однією роллю та без певних взаємодій між ними.

Цей механізм дозволяє включити у дизайн ролі допоміжного персоналу, і навіть ролі слухачів. Оскільки Learning Design відокремлює діяльність від структур діяльності, а ті – від ролей та ресурсів, всі вони стають компонентами багаторазового використання. Вони поєднуються під поняттям Метод, який використовує знайому структуру п'єси з актами та ролями в кожному акті.

Відповідність між термінологією базової моделі Learning Design та термінологією мета-моделі простору оцінки продемонстровано в таблиці 2.1

Таблиця 2.1 – Відповідність сутностей

Мета-модель оцінювального простору для професійного тренінгу	Основна концепція дизайну навчання
Activity, project; QUIZ and evaluation	Activity
Task, and subtask of project or activity	Activity structure
Prerequisite	Prerequisite
The phase of activity, project, QUIZ, or evaluation	Play
Members, and team	Person
Coordinator, teacher, mentor and tutor	Staff
Learner	Learner
Activity, project, QUIZ, or evaluation Objective	Learning objective
Course and Chapter	Learning Object
Tools	Services
Evaluation space	Environment

Певні сервіси також вказані у специфікації навчального дизайну. Служба надає спільну функцію, таку як електронна пошта, конференц-зв'язок або служба оголошень, яка не може бути визначена за допомогою URL під час проектування, але натомість може бути прив'язана системою виконання лише після того, як навчальний проект буде створено та реальні люди будуть призначені різні ролі.

2.4 Алгоритм для генерації та оцінки тестів для тренінгу

Запропоновано алгоритм інтелектуальної системи генерації та оцінки тестів тренінгу з різними рівнями складності представлений на рисунку 2.2.

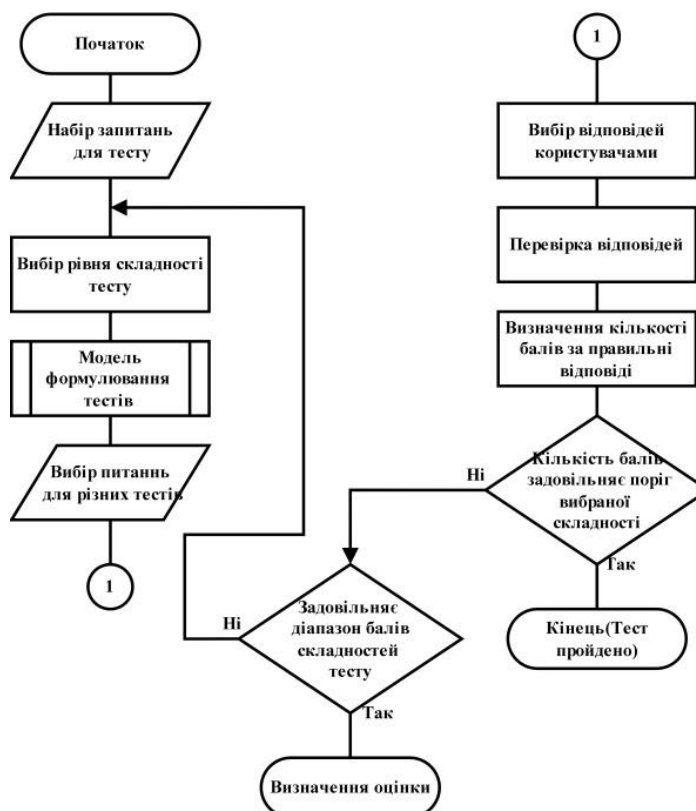


Рисунок 2.2 – Граф-схема алгоритму для генерації та оцінки тестів

Для створення тестів для тренінгу потрібна наявність вже сформульованих питань. Існують вимоги до набору питань, кожне із них має бути оцінено за рівнем складності.

Коли ці дві вимоги задоволені, запропонована інтелектуальна система може

бути застосована для генерації та розташування у послідовному порядку питань для тестування електронного навчання, що використовується для оцінювання. Набір питань та їх складність може бути підготовлена заздалегідь в окремому модулі. Рівень складності питань визначається викладачем, що готує тест [5].

Другий етап цієї системи дозволяє генерувати тести з різним рівнем складності. Це можна зробити шляхом вибору рівня складності тесту. Можна сформувати кілька рівнів складності, які відповідають різним ступеням складності тесту. Немає жодних перешкод для того, щоб диференціювати стільки рівнів сформованих тестів, скільки потрібно.

Наступним етапом даного алгоритму є формулювання математичної моделі для відбору питань, з яких складаються конкретні тести. Ця математична модель повинна гарантувати, що вибрані питання не повторюватимуться у згенерованому тесті. Це означає, що найкраще підходить окремий випадок методу цілочисельного лінійного програмування.

Коли питання, що становлять рівень тесту, відомі, користувач може дати свої відповіді. Після того, як відповіді дано, система перевіряє, на скільки питань дано правильну відповідь. Наступним етапом є визначення балів за тест залежно від кількості правильних відповідей на запитання. Сума балів правильних відповідей обчислюється з урахуванням складності питань, визначеної першому етапі запропонованого алгоритму, і правильних відповідей. Потім необхідно перевірити, чи обчислена сума балів відповідає кількості правильних відповідей для необхідної кількості балів для обраного рівня тесту. Якщо це так, то алгоритм завершується, але якщо це не так, необхідно провести верифікацію. У ході верифікації перевіряється, чи отримані бали з тесту знаходяться в межах допустимого діапазону для іншого рівня складності тесту. Якщо це так, алгоритм завершується з пропозицією відповідної оцінки. Якщо це не так, алгоритм запропонованої інтелектуальної системи дозволяє користувачеві вибрати новий тест із меншою складністю та провести повторне тестування. Такий сценарій можливий лише в тому випадку, якщо час, що відведений на тестування, ще не закінчився.

2.5 Математична модель на основі цілочисельного програмування для генерації та оцінки тестів професійного тренінгу

Попередньо визначений набір запитань, що стосується конкретної області, позначається $Q \in \{q_1, q_2; \dots, q_M\}$. Цей набір складається з M кількості різних питань.

Кожне запитання має свій власний ступінь складності, який приймає значення в діапазоні $1 \leq d_i \leq N$, де найменший ступінь складності питань дорівнює 1, а значення, що відповідає найвищому ступеню складності, виражається N . Слід зазначити що немає зв'язку між загальною кількістю питань (Q) і ступенями складності.

Вхідні дані для формулювання математичної моделі для генерування та оцінювання тестів електронного навчання наведені в таблиці 2.2.

Таблиця 2.2 – Вхідні дані для формування математичної моделі для генерування та оцінювання тестів професійного тренінгу

Набір запитань	Складність запитань	Двійкові змінні рішення	Рівні складності тестів
q_1	d_1	x_1	L^1, L^2, \dots, L^K
...	
q_i	d_i	x_i	
...	
q_M	d_M	x_M	

З урахуванням описаних вище вихідних даних сформульовано наступну узагальнену інтегральну модель оптимізації лінійного програмування для вибору питань при створенні тестів електронного навчання:

$$\text{minimize } \sum_{i=1}^m x_i = K^k \quad (2.1)$$

В залежності від

$$\sum_{i=1}^m x_i d_i = L^k \quad (2.2)$$

$$L^k \leq L_{max}^k \quad (2.3)$$

$$L^k \geq L_{min}^k \quad (2.4)$$

$$x_i \in \{0,1\} \quad (2.5)$$

Цільова функція (2.1) спрямована на виявлення якнайменшої кількості питань з урахуванням їх складності, задовольняючи при цьому певний рівень тесту. Рівняння (2.2) визначає складність рівня тесту як суму відібраних питань щодо змінних (x_i) та відповідної їм складності (d_i). Обмеження (2.3) і (2.4) визначають нижню L_k min і верхню L_k max межі кількості балів, необхідних для формування обраного ступеня складності тесту. k -й індекс у цих межах означає рівні складності тестів. Двійкові цілочисленні змінні x_i , що виражаються рівнянням (2.5), використовуються для визначення питань, з яких складатиметься конкретний тест. Використання бінарних змінних гарантує, що питання у згенерованому тесті не повторюватимуться. Сформульована вище модель оптимізації є окремим випадком цілочисленного лінійного програмування, в якому змінні (x_i) обмежені цілими числами і невідомі.

Якщо для складання тесту необхідно якомога більше запитань, запропоновану вище математичну модель (2.1)–(2.5) слід модифікувати. У цьому випадку правильна математична модель матиме таку максимізуючу цільову функцію:

$$\text{maximize } \sum_{i=1}^m x_i = K^k \quad (2.6)$$

Відповідно до (2.2)–(2.5)

Математичне формулювання з цільовою функцією (2.6), що відповідає (2.2)–(2.5), визначить якомога більше запитань з урахуванням рівнів складності тесту. Основна ідея цього сценарію — вибрати більшу кількість питань, але з меншим ступенем складності. Використання моделі (2.1)–(2.5) створить тести з меншою кількістю запитань із більшою складністю, тоді як друга модель (2.6), що відповідає (2.2)–(2.5), створить тести з більшою кількістю запитань із меншими труднощами. Обидві моделі можуть бути застосовані для перевірки набутих слухачами знань відповідно до необхідного рівня складності тесту.

Запропоновані математичні моделі, основані на цілісному програмуванні, дозволяють генерувати та оцінювати тести електронного навчання з різним рівнем складності. Перша модель з цільовою функцією (2.1), що підкоряється (2.2)–(2.5), може бути використана, коли необхідно визначити меншу кількість питань, але з більшим ступенем складності, а друга модель з цільовою функцією (2.6), що підкоряється (2.2)–(2.5), призначена для визначення якомога більшої кількості питань, але з меншим ступенем складності.

2.6 Верифікація алгоритму для генерації та оцінки тестів професійного тренінгу

Для верифікації застосування запропонованого алгоритму інтелектуальної системи для генерації та оцінки тестів професійного тренінгу використовується конкретний приклад. Чисельне застосування запропонованих двох моделей оптимізації ілюструється використанням обмеженого числа питань, що дорівнює

двадцяти. Діапазон їхньої складності становить від 1 до 10 (тобто $1 \leq d_i \leq 10$), де більше значення означає більш високий рівень складності.

Використані вихідні дані про складність питань та отримані результати для бінарних цілих змінних (x_i) для трьох різних рівнів складності тесту представлені в таблиці 2.3.

Таблиця 2.3 – Вхідний набір запитань для верифікації алгоритму

Номер питання	Складність питання	$minimize \sum_{i=1}^m x_i$			$maximize \sum_{i=1}^m x_i$		
		L1	L2	L3	L1	L2	L3
1	5	0	1	1	1	1	0
2	10	1	1	1	1	1	1
3	10	1	1	1	0	0	1
4	9	1	1	1	0	1	1
5	7	1	0	0	1	1	1
6	6	0	0	0	1	1	0
7	3	0	0	0	1	1	1
8	5	0	1	1	1	1	1
9	8	1	1	1	1	1	0
10	9	1	0	0	0	1	1
11	6	0	1	1	1	1	0
12	7	1	1	1	1	1	1
13	8	1	0	0	1	0	0
14	3	0	1	1	1	0	1
15	10	1	0	0	1	0	1
16	5	0	1	0	1	1	1
17	7	0	1	1	1	1	1
18	10	1	1	1	0	1	1
19	9	1	1	1	0	1	1
20	8	1	0	0	1	0	0

Загальна оцінка для всіх питань у цьому уривку дорівнює 142. Чисельне тестування проводиться на трьох різних рівнях складності тестів із попередньо визначеними верхніми та нижніми межами для кожного рівня тесту. Кількість вибраних питань і відповідні бали для кожного рівня тесту з використанням обох

сформульованих оптимізаційних моделей з функціями мінімізації та максимізації наведені в таблиці 2.4

Таблиця 2.4 – Результати верифікації моделей

	Верхня границя	Нижня границя	Кількість балів	Кількість запитань
$minimize \sum_{i=1}^m x_i$				
L1	100	90	95	11
L2	89	75	81	9
L3	74	60	62	7
$maximize \sum_{i=1}^m x_i$				
L1	100	90	95	16
L2	89	75	88	14
L3	74	60	74	13

Для самого високого рівня складності тесту (L1) алгоритм підібрав 11 питань. Сума труднощів цих питань знаходиться у визначених границях (табл. 2.4) і рівно 95 балів.

Графічна візуалізація між вибраною кількістю питань і відповідними тестовими балами, в залежності від обраної моделі, представлені на рисунках 2.3 та 2.4.

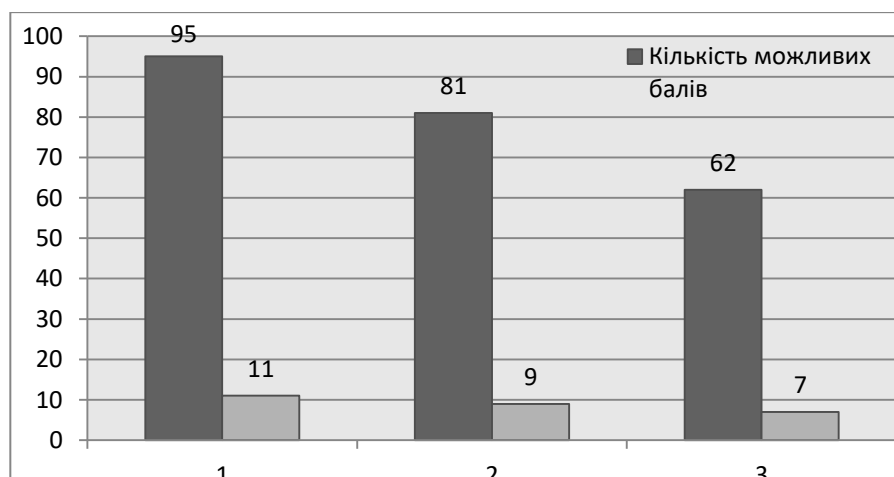


Рисунок 2.3 – Діаграма для мінімізуючої моделі вибору питань

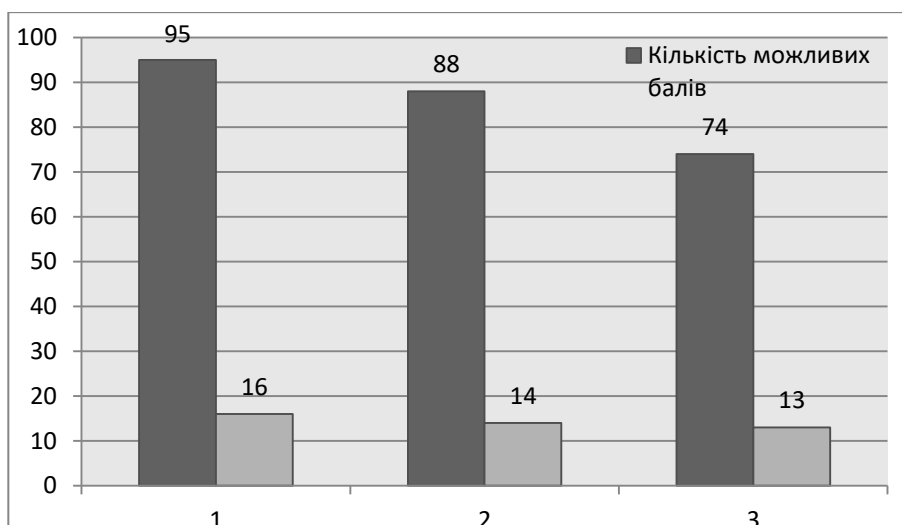


Рисунок 2.4 – Діаграма для максимізуючої моделі вибору питань

Створення тесту для середнього рівня (L2) вимагає зменшення тестового бала. Результати розв'язання задач оптимізації відповідають очікуванням. Вибрана кількість питань дорівнює 9, а загальні бали за питання дорівнюють 81.

Результати генерації тесту з нижчим рівнем складності (L3) показують, що тест складається з 7 питань з балами складності 62. Питання, відібрані розробленою системою рівня L3, продемонстровано на рисунку 2.3.

Можна зазначити, що отримані бали для згенерованих тестів близькі до нижньої межі кожного рівня тесту. Наприклад, для тесту з рівнем L1 бали становлять 95, тоді як нижня межа встановлена на рівні 90. Для рівня L2 згенерованого тесту бали становлять 81, а нижня межа також 75, а для тесту з рівнем L3 бали становлять 62 при нижньому кордоні 60

Використання другої моделі з об'єктивною функцією (2.6) також перевіряється для трьох різних випадків генерації тестів. Мета цієї моделі - визначити якнайбільше питань із меншим ступенем складності. Отримані результати цієї моделі для різних випадків складності тестів показані на рис. 2.4.

Для найвищого рівня (L1) певна кількість питань дорівнює 16, при цьому сума їх труднощів близька до такої в першій моделі, а бал дорівнює 95.

При генерації тесту середнього рівня (L2) вирішення оптимізаційної задачі визначає 14 питань та 88 балів за них.

Результати генерації тесту з нижчим рівнем складності (L3) показують, що тест складається з 13 питань з балами від їх складності 74.

Чисельне застосування другої моделі з цільовою функцією (2.6) за умови (2.2)-(2.5) показує, що отримані бали для згенерованих тестів близькі до верхньої межі кожного рівня тесту. Наприклад, при генерації тесту з рівнем L2 отримано 88 бали при верхньому кордоні 89, а для тесту з рівнем L3 - 74 балів при верхньому кордоні 74. Оцінка 95 для тесту з рівнем L1 знаходиться посередині між нижньою та верхньою межами

2.7 Висновки до другого розділу

В цьому розділі були зазначені основні задачі що до кваліфікаційної роботи магістра. Завдяки чому була зазначена мета роботи, її актуальність та доцільність виконання. Розглянуто переваги та недоліки існуючого методу тестування, та запропоновано алгоритм генерації та оцінювання тестів для того щоб нівелювати недоліки стосовно затрат часу на створення тестів викладачами. Запропоновано та перевірено модель для генерації та оцінювання тестів, за допомогою математичного моделювання, зазначено необхідність практичного моделювання системи.

3 ВИЗНАЧЕННЯ АРХІТЕКТУРИ СИСТЕМИ

3.1 Дослідження загальної архітектури системи

На рисунку 3.1 представлена діаграма прецедентів розробленої системи.

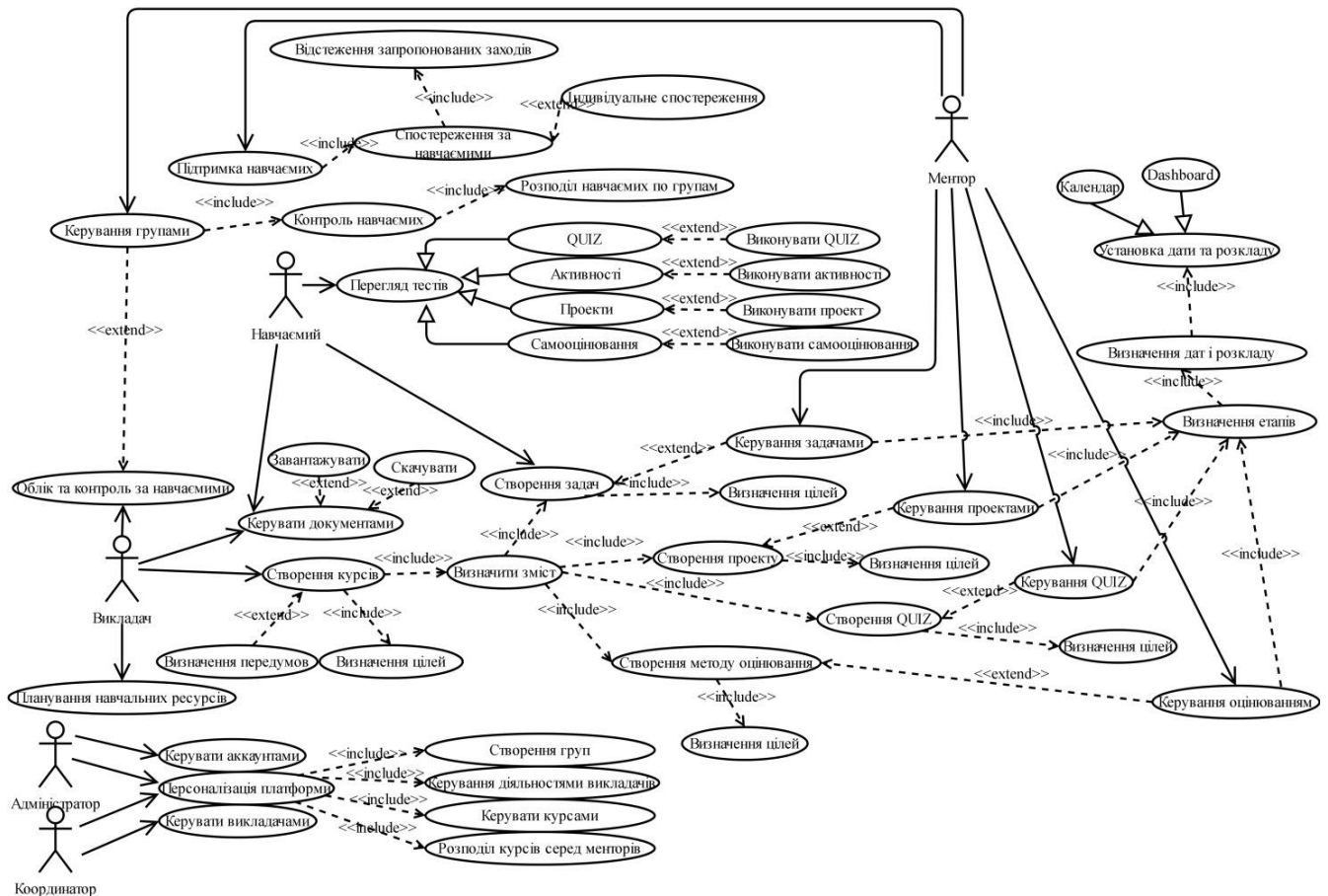


Рисунок 3.1 – Діаграма прецедентів

Виконаємо верхні оцінки обчислювальної складності діаграми за формулами:[23]

$$C_{UML} = a \cdot (n_e + n_e^2 + 3 \cdot n_r)$$

$$C_{path} = a \cdot \left(\frac{n_e^2}{2} + n_r \right)$$

$$\Delta C = a \cdot \left(n_e + \frac{n_e^2}{2} + 2 \cdot n_r \right)$$

де n_e – кількість класифікаторів сутностей діаграми,

n_e^2 – кількість кліток квадратної матриці можливих відносин між сутностями,

n_r – кількість призначених відносин діаграми, множник «3» значить розгляд як самого відношення r , так і двох інцидентних йому сутностей $e1$ і $e2$,

a – це коефіцієнт абстрактності рівня (для технічного завдання, об'єктнокомпонентного, структурно-функціонального, подійно-часового рівнів – 3, для автоматно-алгоритмічного – 2 або 1).

Остаточню:

$$\begin{aligned} C_{UML} &= a \cdot (n_e + n_e^2 + 3 \cdot n_r) = \\ &= 3 \cdot (54 + 54^2 + 3 \cdot 59) = 9441 \end{aligned}$$

$$\begin{aligned} C_{path} &= a \cdot \left(\frac{n_e^2}{2} + n_r \right) = \\ &= 3 \cdot \left(\frac{54^2}{2} + 59 \right) = 4551 \end{aligned}$$

$$\begin{aligned} \Delta C &= a \cdot \left(n_e + \frac{n_e^2}{2} + 2 \cdot n_r \right) = \\ &= 3 \cdot \left(54 + \frac{54^2}{2} + 2 \cdot 59 \right) = 4890 \end{aligned}$$

У результаті зменшення обчислювальної складності синтезу діаграми при верифікації отримано 4890 умовних одиниць аналізу.

Система передбачає 5 основних типи акторів: ментор, навчаємий, викладач, адміністратор та координатор. Навчаємий може переглядати курси, активності, вікторини, проекти, оцінки їх цілі та передумови в будь-який момент, може виконувати тести, завдання, необхідне проектування, завантажувати та скачувати документи. Ментор може створювати, редагувати та видаляти навчаємих та групи, може запрошувати навчаємих до груп, визначати завдання, встановлювати та редагувати дати початку та закінчення завдань, призначати завдання навчаємих. Викладач може створювати завдання, вікторини, проекти для курсів зі своїми поправками, також він планує план навчання тренінгу, може скачувати та

завантажувати документи. Адміністратор може призначати курси менторам відповідно до спеціалізації кожної людини, може керувати курсом, може керувати правами доступу вчителів і навчаємих. Координатор може створювати, редагувати та видаляти вчителів, відстежувати їх активність, створювати групи, може призначати курси менторам відповідно до спеціалізації кожної людини, може керувати курсом.

Діаграма компонентів розроблювальної системи представлена на рисунку 3.2

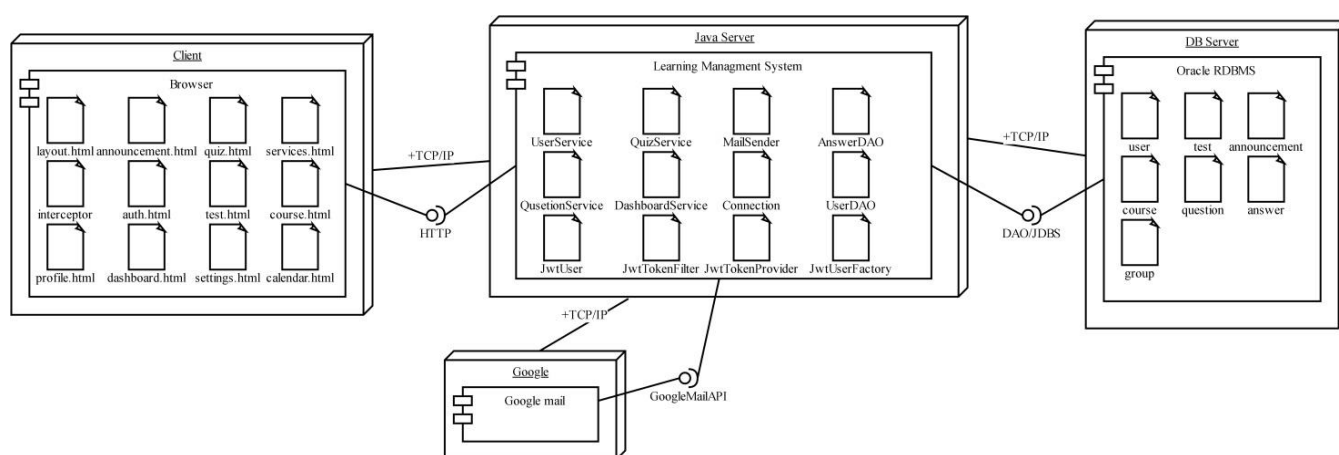


Рисунок 3.2 – Діаграма компонентів

Інтелектуальна система складається з трьох основних компонентів: «DB Server» – сервер з базою даних від компанії Oracle, яка зберігає у собі всю необхідну інформацію для функціонування сервісу, «Java Server» – сервер, який забезпечує обробку запитів клієнта, та формує відповіді, має зв'язок з базою даних, та за потреби звертається до неї за потрібною інформацією, «Client» – клієнтська частина сервісу, забезпечує комунікацію з клієнтом, містить графічний інтерфейс користувача, виконує валідацію форм, які заповнює користувач.

Також на рисунку представлений додатковий компонент «Google», котрий забезпечує роботу з відправкою повідомлень на пошту.

Компонент «DB Server» містить такі програмні компоненти: user, test, quiz, group, course, file, activity, answer, announcement які є таблицями у реляційній базі даних Oracle RDBMS.

Компонент «Java Server» містить такі програмні компоненти: UserService, QuizService, QuestionService, JwtUser, JwtTokenFilter, JwtTokenProvider, JwtUserFactory, UserDao, AnswerDao, MailSender, DashboardService, Connection які є класовими файлами програми.

Компонент «Client» містить такі програмні компоненти: layout, announcement, quiz, test, interceptor, profile, dashboard, calendar, settings, course, services, auth які є файлами, що містять графічний інтерфейс користувача та обробники запитів користувача.

3.2 Дослідження динамічної архітектури системи

Діаграма послідовностей відправки HTTP-запитів та отримання HTTP-відповідей між користувачем, браузером(клієнтом), сервером та базою даних під час реєстрації та авторизації в системі представлена на рисунку 3.3.

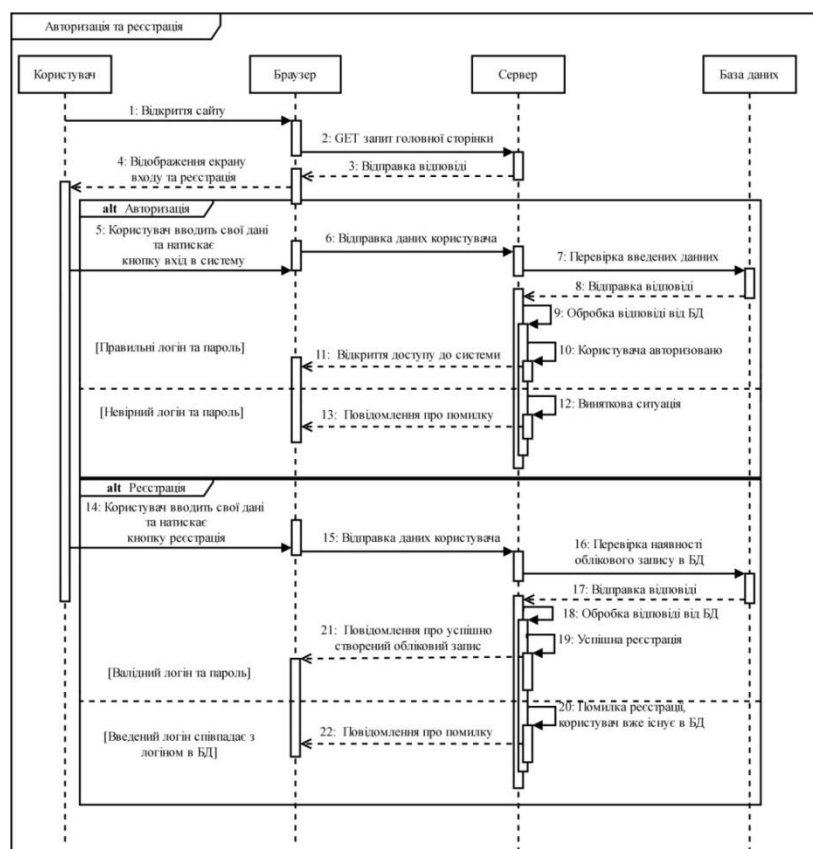


Рисунок 3.3 – Діаграма послідовності авторизації та реєстрації

З рисунку 3.3 видно, що для авторизації користувача спочатку необхідно отримати форму для відображення екрану входу та реєстрації, тому клієнт(браузер) відправляє запит до серверної частини для отримання форми екрану входу та реєстрації. Сервер обробляє цей запит, формує відповідь у вигляді екрану входу та реєстрації та відправляє її на браузер, котрий в свою чергу відображає форму у користувацькому інтерфейсі.

Для того щоб авторизуватись в системі користувачу потрібно ввести логін, пароль та натиснути кнопку «Вхід в систему», після цього браузер відправляє введені дані користувача на сервер, котрий в свою чергу перевіряє правильність введених даних користувачем на сервері бази даних, формується відповідь для серверу. Сервер, обробляє відповідь від бази даних. Якщо коректно введено логін та пароль користувачем – сервер відкриває доступ до системи, та надає відповідні можливості користувачу з огляду на його роль в системі (навчасий, ментор, викладач, адміністратор). Якщо логін та пароль невірно введено, тоді сервер відправляє відповідне повідомлення про виняткову ситуацію клієнту.

Для того щоб зареєструватись в системі користувачу потрібно ввести логін, пароль та натиснути кнопку «Реєстрація», після цього браузер відправляє введені дані користувача на сервер, котрий в свою чергу перевіряє на їх дублікати в базі даних, формується відповідь для серверу. Сервер, обробляє відповідь від бази даних. Якщо логін та пароль – валідні та в базі даних дублікатів немає, створюється новий обліковий запис та відкривається доступ до системи. Якщо логін та пароль некоректно введено або в базі даних існують дублікат, тоді сервер відправляє відповідне повідомлення клієнту про виняткову ситуацію клієнту.

Виконаємо верхню оцінку обчислювальної складності:

$$C_{UML} = a \cdot (n_e + n_e^2 + 3 \cdot n_r) =$$

$$= 3 \cdot (26 + 26^2 + 3 \cdot 22) = 2304$$

$$C_{path} = a \cdot \left(\frac{n_e^2}{2} + n_r \right) =$$

$$= 3 \cdot \left(\frac{26^2}{2} + 22 \right) = 1080$$

$$\Delta C = a \cdot \left(n_e + \frac{n_e^2}{2} + 2 \cdot n_r \right) =$$

$$= 3 \cdot \left(26 + \frac{26^2}{2} + 2 \cdot 22 \right) = 1224$$

У результаті зменшення обчислювальної складності синтезу діаграми при верифікації отримано 1224 умовних одиниць аналізу.

Діаграма послідовностей для проходження тренінгу та тестування навчаємим, представлена на рисунку 3.4.

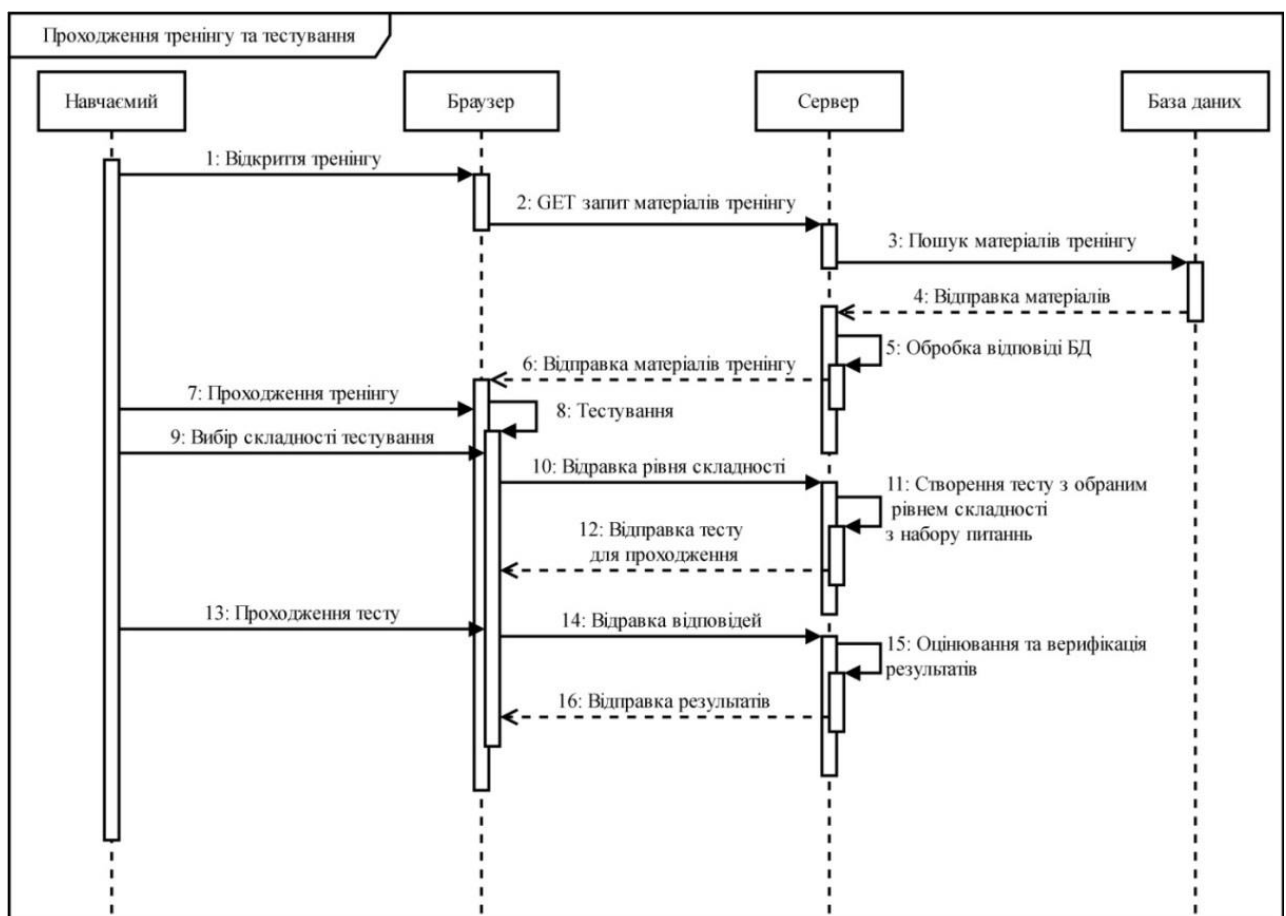


Рисунок 3.4 – Діаграма послідовності для проходження тренінгу та тестування

З рисунку 3.4 видно, що для того щоб пройти тренінг, навчаємому спочатку необхідно вибрати необхідний тренінг із запропонованих. Після того як тренінг вибрано, браузер відправляє GET запит на сервер, з метою отримати матеріали

тренінгу з бази даних. Сервер обробляє цей запит, та отримує необхідні матеріали з бази даних, разом з набором питань та відповідей для майбутнього тесту. Сервер відправляє необхідні дані для користування навчаємому в браузер. Навчаємий в свою чергу проходить необхідні матеріали тренінгу(лекції, презентації, відеозаписи, тощо).

Після проходження теоретичної частини, навчаємому необхідно пройти тестування, що то матеріалів тренінгу. Тест вважається зарахованим в тому випадку коли оцінка за його проходження задовольняє необхідний бал котрий призначається викладачем або ментором. Перед тим як розпочати тестування навчаємий обирає рівень складності тесту, а потім він передається серверу. В свою чергу сервер створює за обраним рівнем складності тест з набору питань котрі в нього присутні. В залежності від вибраної складності тесту:

– Тест може складатись з великої кількості запитань, але відповіді за ці питання приносять незначну кількість балів.

– Тест складається з меншої кількості запитань, та відповіді за питання приносять більшу кількість балів.

Від рівня складності тесту залежить лише кількість можливих питань та кількість балів за правильні відповіді. Сервер відправляє зібраний тест на браузер, для проходження навчаємым. Після того як всі відповіді виставленні, тест вважається завершеним, та він передається для оцінювання на сервер. Сервер в свою чергу отримує тест, та звіряє відповіді навчаємого з правильними відповідями котрі в нього наявні. Підраховується кількість балів за відповіді, та виставляється оцінка за тест, котра відразу приходить на браузер для перегляду. Якщо оцінка задовольняє встановлені критерії, тоді тест вважається пройденим успішно. В іншому випадку навчаємому пропонується знизити складність тесту та пере пройти його.

Виконаємо верхню оцінку обчислювальної складності:

$$C_{UML} = a \cdot (n_e + n_e^2 + 3 \cdot n_r) =$$

$$= 3 \cdot (20 + 20^2 + 3 \cdot 16) = 1404$$

$$\begin{aligned}C_{path} &= a \cdot \left(\frac{n_e^2}{2} + n_r \right) = \\ &= 3 \cdot \left(\frac{20^2}{2} + 16 \right) = 648 \\ \Delta C &= a \cdot \left(n_e + \frac{n_e^2}{2} + 2 \cdot n_r \right) = \\ &= 3 \cdot \left(20 + \frac{20^2}{2} + 2 \cdot 16 \right) = 756\end{aligned}$$

У результаті зменшення обчислювальної складності синтезу діаграми при верифікації отримано 756 умовних одиниць аналізу.

3.3 Висновки до третього розділу

В цьому розділі розглянуто статичну та динамічну архітектуру розроблювального прототипу. Продемонстровано та описано діаграму варіантів використання та діаграму компонентів системи. Побудовано та описано дві діаграми послідовності: для автентифікації та проходження тестування після тренінгу. Розраховано верхні оцінки обчислювальної складності діаграм.

4 РЕАЛІЗАЦІЯ ТА ВЕРИФІКАЦІЯ

4.1 Проектування серверної та клієнтської частини системи

Розроблювана система має дволанкову клієнт-серверну архітектуру. Це означає, що клієнтська частина функціоналу системи виконується безпосередньо на пристрої користувача. Серверна частина, в свою чергу, відповідає за обробку та зберігання даних сервісу. Передача даних між серверною та клієнтською частиною відбувається за допомогою протоколу «HTTP».

Серверна частина розроблюваного сервісу поділяється на наступні модулі:

- «Java Virtual Machine» – віртуальна машина Java - основна частина виконуючої системи Java, так званої Java Runtime Environment. Віртуальна машина Java виконує байт-код Java, попередньо створений із вихідного тексту Java-програми компілятором Java.

- «Spring» – це середовище на основі Java з відкритим вихідним кодом, що використовується для створення мікросервісу;

- «Spring Security» – надає механізми побудови систем автентифікації та авторизації, а також інші можливості безпеки для корпоративних додатків, створених за допомогою Spring Framework та JSON Web Token (JWT).

- Модуль автентифікації;

- Модуль керування даними;

- Модуль керування оголошеннями;

- Модуль керування пошуку;

- Модуль генерації тестів;

- База даних Oracle Database – об'єктно-реляційна система управління базами даних від компанії Oracle.

Загальна структура серверної частини сервісу зі всіма зв'язками між її модулями представлена на рисунку 4.1.

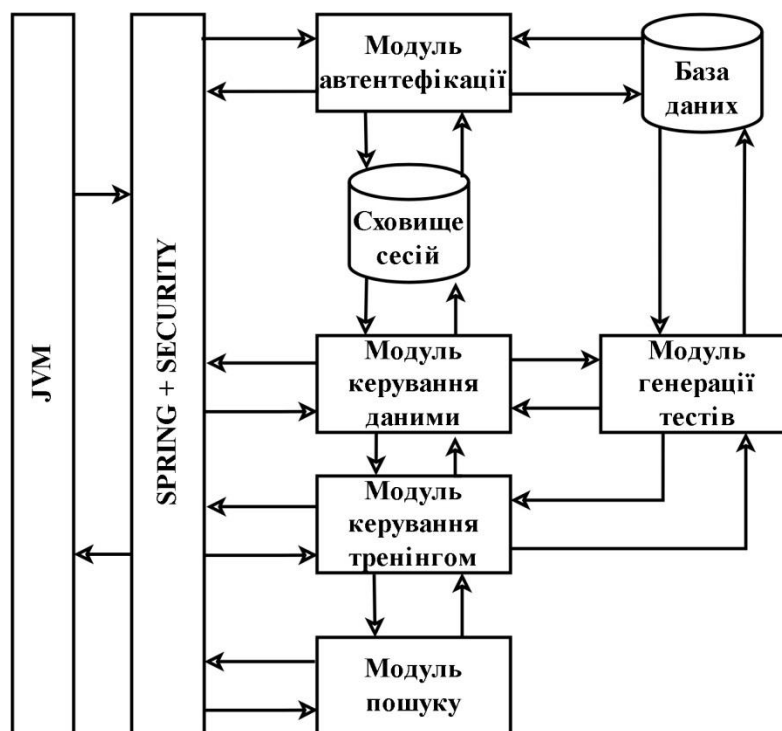


Рисунок 4.1 – Структура серверної частини системи

Модуль автентифікації – модуль, який відповідає за автентифікацію користувачів. Модуль має доступ до бази даних, у якій зберігається необхідна для автентифікації користувачів інформація. Після проведення автентифікації модуль передає дані користувача у сховище сесій, в якому відбувається зберігання сесій користувачів.

Модуль керування даними – відповідає за запис у базу даних користувацької інформації під час реєстрації користувачів та зчитування з бази даних необхідної інформації про користувачів під час роботи системи. Також має доступ до сховища сесій користувачів – це потрібно для перевірки автентифікації користувачів.

Модуль керування тренінгом – відповідає за створення та перегляд користувачем оголошень. Також відповідає за додавання викладачами тренінгів та коментарів до оголошень й їх перегляд навчаємими.

Модуль пошуку – відповідає за пошук тренінгів у базі даних згідно з заданими користувачем параметрами та фільтрами. Має доступ до бази даних, з якої зчитує дані, та передає модулю керування даними тренінгів.

Модуль генерації тестів – відповідає за створення та проходження тесту після тренінгу, згідно з складністю та моделлю обраними навчаємим. Має доступ до бази даних, з якої зчитує необхідні питання стосовно обраного тренінгу, з котрих і буде складатися тест.

Загальна структура клієнтської частини сервісу зі всіма зв'язками між її модулями представлена на рисунку 4.2.

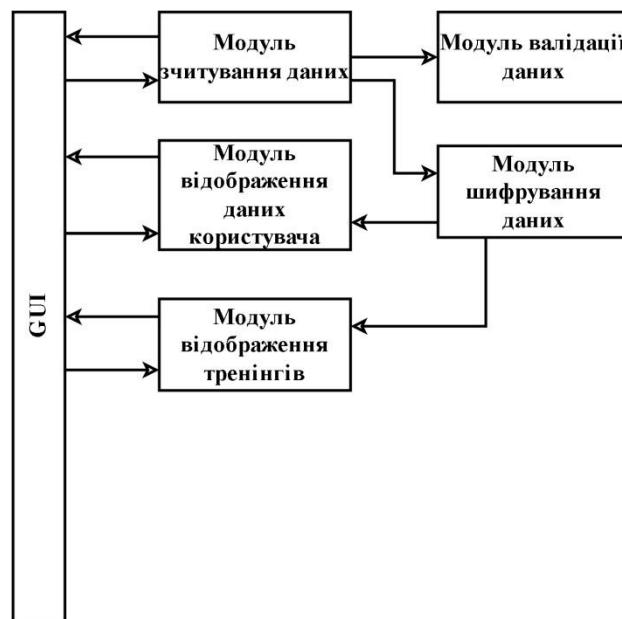


Рисунок 4.2 – Структура клієнтської частини системи

Клієнтська частина розроблювальної системи складається з наступних модулів:

- GUI – користувацький інтерфейс користувача;
- Модуль зчитування даних;
- Модуль валідації даних;
- Модуль відображення даних користувачів;
- Модуль відображення тренінгів;
- Модуль шифрування даних.

Користувацький інтерфейс відповідає за чітке та логічне виведення інформації й відображення форм для введення інформації користувачем.

Модуль зчитування даних, введених користувачем виконує зчитування даних, які вводить користувач у форми у користувацькому інтерфейсі. Далі модуль передає зчитану інформацію до модулю валідації даних.

Модуль валідації даних, введених користувачем, відповідає за перевірку правильності введених користувачем даних. Неправильно заповнені форми у користувацькому інтерфейсі можуть викликати помилку при їх обробці на сервері, тому перед відправкою цих даних до серверу необхідно виконати їх перевірку. Потім модуль перевірки даних передає вже перевірені дані до серверу за допомогою HTTPS-запиту та JWT.

Модуль відображення даних користувачів відповідає за підготовку даних користувачів перед відправкою їх до користувацького інтерфейсу.

Модуль відображення тренінгів виконує підготовку даних тренінгів перед відображенням їх у користувацькому інтерфейсі.

Модуль шифрування даних, відповідає за безпечне передавання даних з допомогою Spring Security та JWT по протоколу HTTPS.

Загальна структура розробленої системи представлена на рисунку 4.3.

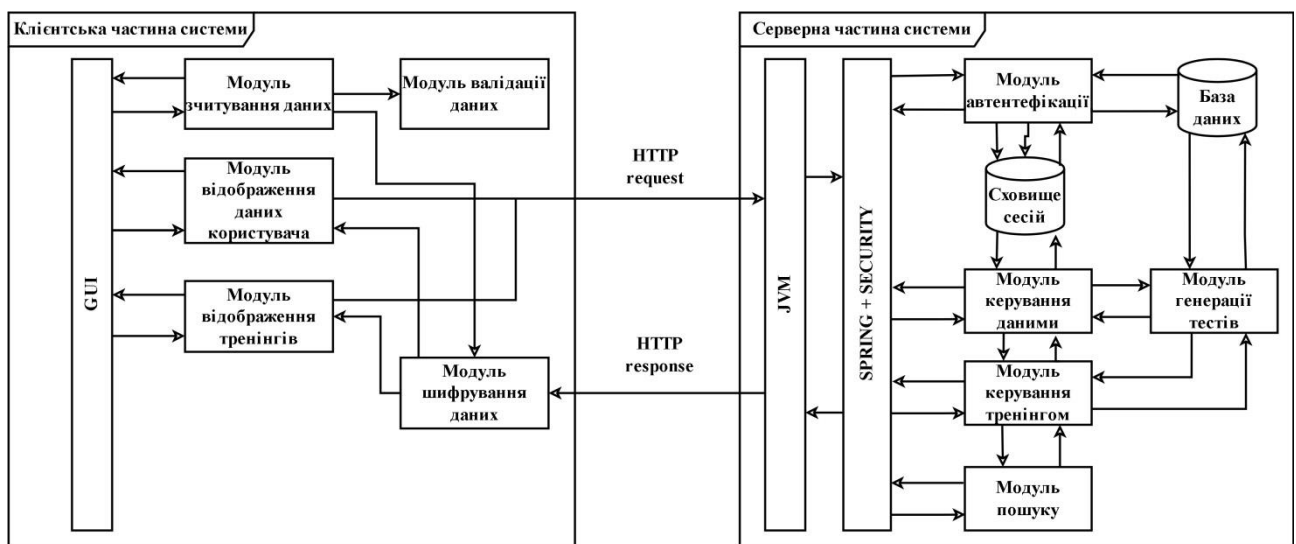


Рисунок 4.3 – Загальна структура розробленої системи

Передача даних між серверною та клієнтською частиною сервісу виконується за допомогою протоколу передачі «HTTPS». На серверній частині

модулі «JVM» та «Spring» відповідає за прийом та передачу даних, в свою чергу, на клієнтській частині за передачу та прийом даних від серверної частини відповідає модуль шифрування даних.

З рисунку видно, що взаємодія між серверною та клієнтською частинами сервісу відбувається за рахунок відправки клієнтською частиною HTTPS-запитів до серверної частини та отримання клієнтською частиною HTTPS-відповідей від серверної частини сервісу.

4.2 Проектування та розробка діаграми класів

Для проектування та розробки системи був використаний паттерн Model View Controller(MVC).

MVC – це паттерн, який включає кілька дрібніших шаблонів. При використанні MVC, на три окремі компоненти розділені модель даних програми, інтерфейс користувача і логіка взаємодії користувача з системою, завдяки чому модифікація одного з цих компонентів надає мінімальний вплив на інші або не робить його зовсім.

Основна мета застосування MVC полягає у розподілі даних та бізнес-логіки від візуалізації. За рахунок такого поділу підвищується можливість повторного використання програмного коду: наприклад, додати уявлення даних будь-якого існуючого маршруту не тільки у вигляді HTML, а й у форматах JSON, XML, PDF, XLSX стає дуже просто і не потребує змін шару бізнес-логіки вихідного маршруту. Також спрощується і супровід програмного коду: внесення змін до зовнішнього вигляду, наприклад, не відбивається на бізнес-логіці, а зміни бізнес-логіки не торкаються візуалізації.

Концепція MVC поділяє дані, представлення та обробку дій користувача на компоненти:

– Модель/Model — є об'єктною моделлю певної предметної області, включає дані та методи роботи з цими даними, реагує на запити з контролера, повертаючи дані та/або змінюючи свій стан. При цьому модель не містить у собі

інформації про способи візуалізації даних або формати їх подання, а також не взаємодіє з користувачем безпосередньо.

– Вигляд/View – відповідає за відображення інформації (візуалізацію). Одні й самі дані можуть представлятися різними способами й у різних форматах.

– Контролер/Controller – забезпечує зв'язок між користувачем та системою, використовує модель та уявлення для реалізації необхідної реакції на дії користувача.

Для з'єднання та обробки інформації з базою даних використовується стандарт Java DataBase(JDBC) Connectivity та Database Access Object(DAO).

JDBC – платформи незалежний промисловий стандарт взаємодії Java-додатків з різними СУБД, реалізований як пакет java.sql, що входить до складу Java SE. JDBC заснований на концепції драйверів, що дозволяють отримувати з'єднання з базою даних за спеціально описаним URL.

У програмному забезпеченні DAO – абстрактний інтерфейс до якогось типу бази даних або до механізму зберігання.

Діаграма класів розробленої системи представлена в додатку А.

4.3 Розробка бази даних

ER-модель (від англ. Entity-Relationship model, модель "сутність - зв'язок") - модель даних, що дозволяє описувати концептуальні схеми предметної області. ER-модель використовується при високорівневому (концептуальному) проектуванні бази даних. З її допомогою можна виділити ключові сутності та позначити зв'язки, які можуть встановлюватись між цими сутностями.

Функції системи зберігання інформації виконує об'єктно-реляційна система управління базами даних компанії Oracle.

Загальна структура бази даних зі зв'язками між таблицями приведена на рисунку 4.4. Розроблену базу даних нормалізовано до третьої нормальної форми.

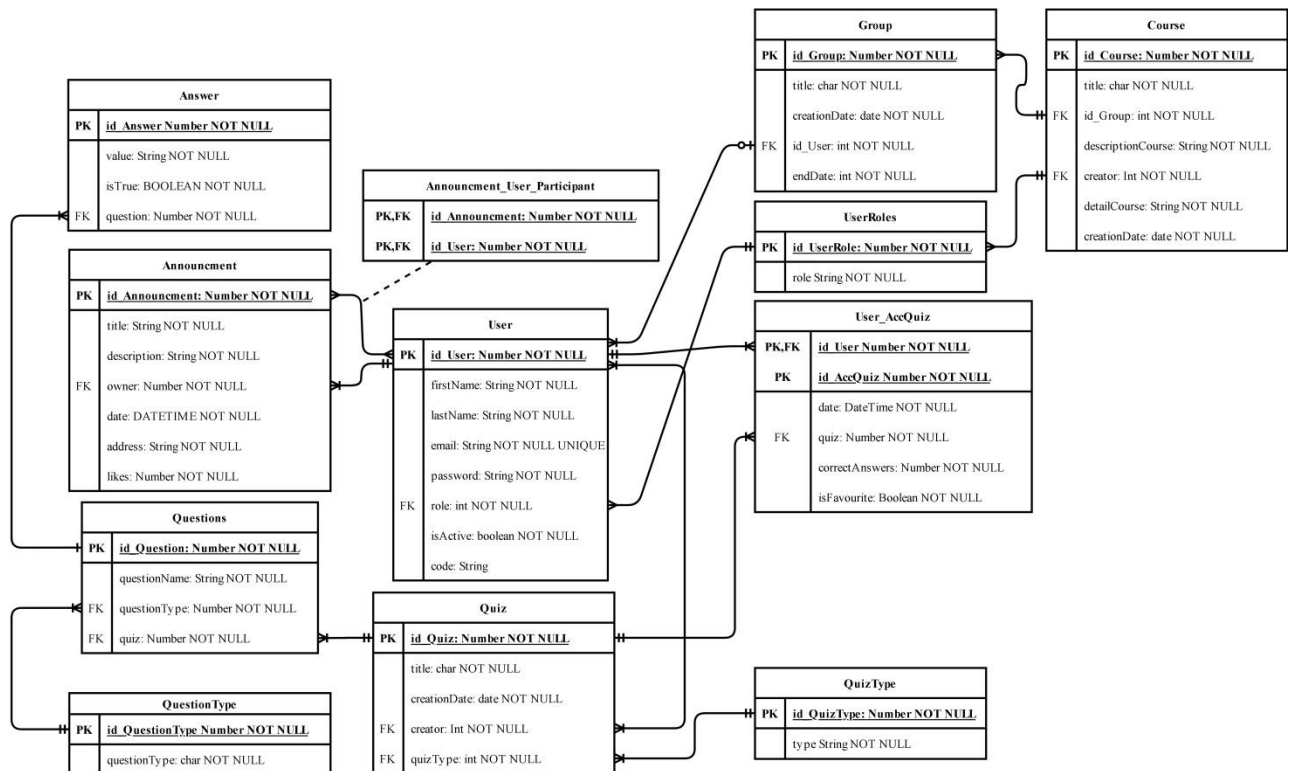


Рисунок 4.4 – Діаграма сутностей бази даних

База даних розробленої системи складається з дванадцяти таблиць:

- Answer – таблиця, у якій зберігаються відповіді на запитання;
- User – таблиця, у якій зберігаються дані користувачів;
- UserRoles – таблиця, у якій зберігаються можливі ролі користувачів;
- Group – таблиця, у якій зберігається інформація про групи користувачів;
- Course – таблиця, у якій зберігається інформація про тренінги;
- Quiz – таблиця, у якій зберігаються тести;
- Questions – таблиця в котрій зберігаються запитання;
- QuestionType – таблиця з можливими типами питань;
- QuizType – таблиця с можливими тренінговими напрямками тестів;
- User_AccQuiz – таблиця с відповідями котрі користувачі встановлювали при проходженні тесту;
- Announcement – таблиця з необхідною інформацією для відправки повідомлень;

– Announcement_User_Participant – асоціативна таблиця для вирішення проблеми з'єднання типу «багато до багатьох».

Таблиця Answer складається з чотирьох полів:

- id_Answer – унікальний номер відповіді (первинний ключ);
- value – строкове значення відповіді;
- isTrue – булеве значення відповіді;
- question – номер запитання(зовнішній ключ).

Таблиця User складається з восьми полів:

- id_User – унікальний номер користувача (первинний ключ);
- firstName – власне ім'я користувача;
- lastName – прізвище користувача;
- email – електронна пошта користувача;
- password – хеш сума паролю користувача;
- role – роль користувача в системі (зовнішній ключ);
- isActive – статус користувача в системі;
- code – код підтвердження, необхідний для підтвердження облікового запису, котрий приходить на поштову адресу (після активації, видаляється).

Таблиця UserRoles складається з двох полів:

- id_UserRole – унікальний номер ролі (первинний ключ);
- role – роль користувача в системі;

Таблиця Group складається з п'яти полів:

- id_Group – унікальний номер групи (первинний ключ);
- title – назва групи;
- creationDate – дата створення;
- id_User – унікальні номери учасників групи(зовнішній ключ);
- endDate – дата розформування групи.

Таблиця Course складається з семи полів:

- id_Course – унікальний номер тренінгу (первинний ключ);

- title – назва тренінгу;
- id_Group – унікальні номери груп котрі входять до тренінгу (зовнішній ключ);
- descriptionCourse – опис тренінгу;
- creator – користувач створивший цей тренінг (зовнішній ключ);
- detailCourse – умови тренінгу;
- creationDate – дата створення тренінгу.

Таблиця Quiz складається з п'яти полів:

- id_Quiz – унікальний номер тесту (первинний ключ);
- title – назва тесту;
- creationDate – дата створення тесту;
- creator – користувач котрий ініціював створення тесту (зовнішній ключ);
- quizType – тип питань в тесті (зовнішній ключ).

Таблиця QuizType складається з двох полів:

- id_QuizType – унікальний номер типу питання (первинний ключ);
- type – назва типу питань.

Таблиця Questions складається з чотирьох полів:

- id_Question – унікальний номер питання (первинний ключ);
- questionName – назва питання;
- questionType – тип питань(зовнішній ключ);
- quiz – номер тесту в котрому це питання зустрічається (зовнішній ключ).

Таблиця QuestionType складається з двох полів:

- id_QuestionType – унікальний номер типу питання(первинний ключ);
- questionType – назва типу питань.

Таблиця User_AccQuiz складається з шести полів:

- id_UserNumber – унікальний номер навчаємого котрий проходив тест(первинний ключ,зовнішній ключ);
- id_AccQuiz_Number – унікальний номер оцінки (первинний ключ);
- date – дата проходження тесту;

- quiz – номер тесту(зовнішній ключ);
- correctAnswers – кількість правильних відповідей;
- isFavourite – булеве значення, котре залежить чи сподобався тест навчаємому котрий його проходив .

Таблиця Announcement складається з семи полів:

- id_Announcement – унікальний номер повідомлення(первинний ключ);
- title – назва повідомлення;
- description – текст повідомлення;
- owner – унікальні номери користувачів котрі отримують повідомлення (зовнішній ключ);
- date – дата відправки;
- address – електронна адреса;
- likes – кількість хвалебних позначок.

Асоціативна таблиця Announcement_User_Participant складається з двох полів:

- id_Announcement – унікальний номер повідомлення;
- id_User – унікальний номер користувача.

4.4 Верифікація розробленої моделі генерації та оцінки тестів на практиці

Для перевірки працездатності розробленої моделі генерації та оцінювання тестів в системі, був створений набір із двадцяти теоретичних питань за аналогією таблиці 2.3. Кожне запитання має свій рівень складності та кількість балів за правильну відповідь на це питання. В залежності від обраної моделі генерації тестів та рівня складності тесту користувачем, системою було створено два тести котрі приведені на рисунках 4.5 та 4.6.

Перший тест було згенеровано с десяти запитань та сумарною кількістю балів за тест 91. В тесті були використанні переважно запитання з рівнями складності: сім, вісім, дев'ять та десять.

Question 2 of 10

return to profile

2 of 10

0 9 0
0 0 0
0 0 0
0

Answer selection

0 9 5 3

previous next

Choose the 1000th number after the decimal point of the π - number

Complexity level 10

Рисунок 4.5 – Тест створений за мінімізуючою моделлю та максимальним рівнем складністю питань

Другий тест було згенеровано с 16 запитань та сумарною кількістю балів за тест 95. В тесті були використанні переважно запитання з рівнями складності: від одного до шести.

Question 1 of 16

return to profile

1 of 16

0 0 0
0 0 0
0 0 0
0 0 0

← →

TRUE/FALSE

0 9 5 3

previous next

2 + 2 = 4

Complexity level 1

True
False

Рисунок 4.6 – Тест створений за максимізуючою моделлю та мінімальним рівнем складністю питань

Проведений експеримент доводить, що використані моделі для генерації та оцінки тестів в системі відповідають очікуваному математичному результату верифікації моделей.(таб. 2.3)

4.5 Порівняння та аналіз результатів прототипу

Після того, як програмний прототип був розроблений, необхідно перевірити досягнення поставленої мети, а саме мінімальне збільшення затрат часу необхідного для обробки сервером, генерації та оцінювання тесту в порівнянні з існуючою системою на базі LMS «Moodle».

Була поставлена гіпотеза, перевірити чи збільшився час необхідний для обробки сервером запитань для створення тесту, для перевірки буде використана метрика – середній час обробки сервером запиту від навчаємого для десяти запитів з різними кількостями запитань у тесті, в порівнянні з існуючою системою на базі LMS «Moodle».

Для цього необхідно провести дослідження, яке полягає у перевірці десяти запитів з різною кількістю запитань, спочатку у прототипі, а потім у аналогу на базі «Moodle». Щоб дослідження мало достовірні данні, необхідно виконувати його на одному й тому самому апаратному обладнанні. Результати порівняння експерименту зображено на таблиці 4.1.

Таблиця 4.1 – Результати порівняння

Номер запиту	Кількість запитань в тесті	Виконання запиту в прототипі, мс.	Виконання запиту в системі на базі «Moodle», мс.	Збільшення часу у відсотках, %
1	≈10	236	219	7,20
2	≈15	240	227	5,41
3	≈20	251	237	5,57
4	≈25	264	245	7,19
5	≈30	269	247	8,17
6	≈35	276	250	9,42
7	≈40	280	252	10,00
8	≈45	290	260	7,96
9	≈50	294	265	9,86
10	≈55	301	271	9,96

Знаючи збільшення часу у відсотках для кожного запиту, можна розрахувати, за допомогою середнього арифметичного, середнє зменшення часу. Після розрахунку отримано значення – 8,07%, що є в межах допустимого, так як генерація тестів відбувається безпосередньо на сервері на відміну від «Moodle», котрий отримує готовий тест з бази даних. Середнє значення обробки запиту в системі на базі «Moodle» – 247,3 мс. Середнє значення обробки запиту прототипу – 270,1 мс

Відобразимо отримані дані з таблиці 4.1 на графіку зображеному на рисунку 4.7.

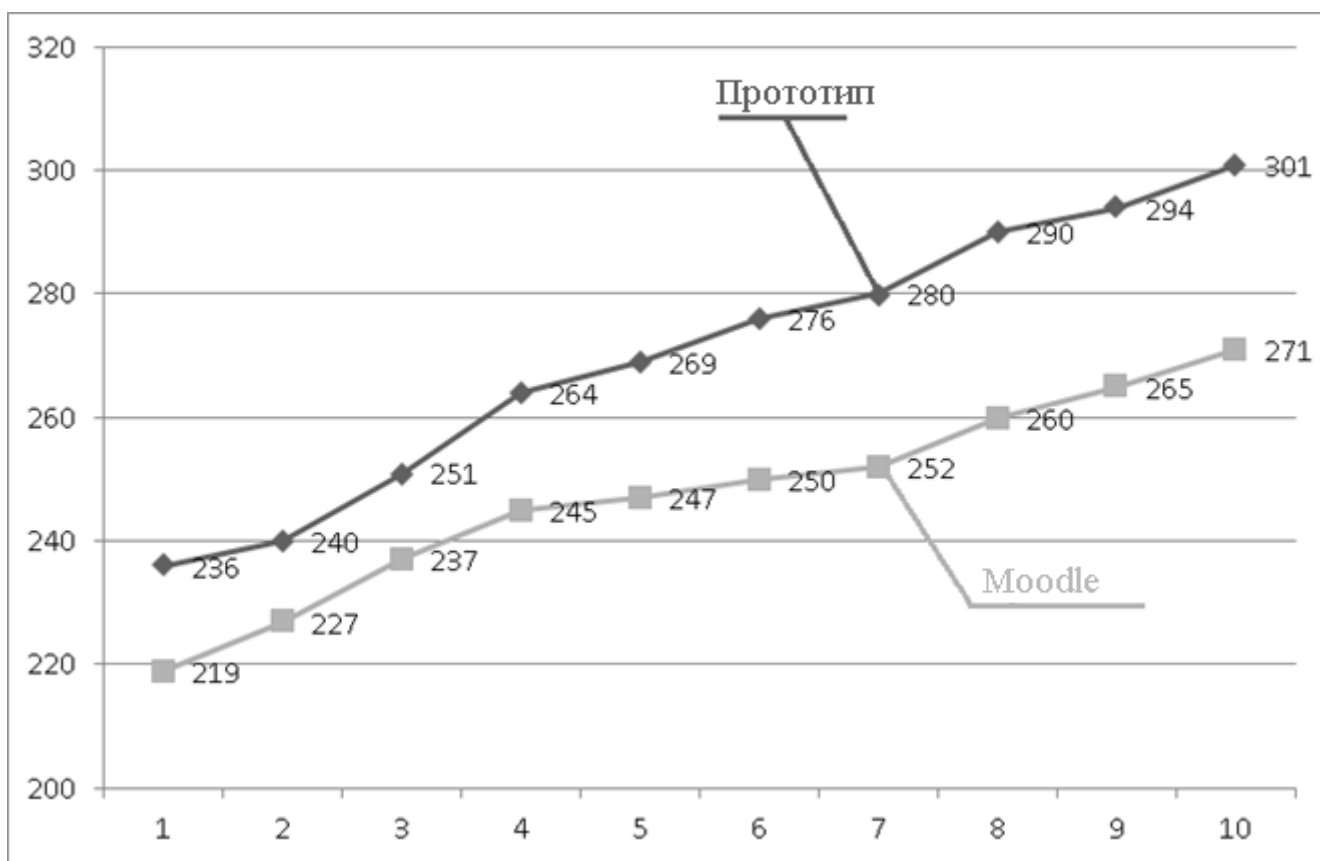


Рисунок 4.7 – Графік порівняння двох систем

З графіку можна побачити лінійне зростання часу обробки запиту сервером як для серверу прототипу так і для серверу аналогу. Збільшення часу обробки запиту на сервері прототипу, нівелюється тим що викладачу не потрібно створювати та оцінювати тести, на відміну від аналогічної системи «Moodle».

4.6 Висновки до четвертого розділу

В цьому розділі описано та продемонстровано загальну структуру, базу даних та програмні класи розробленого прототипу. Для проектування та розробки системи був використаний паттерн MVC мову програмування Java.

Функції системи зберігання інформації виконує об'єктно-реляційна система управління базами даних компанії Oracle. База даних складається з дванадцяти таблиць.

Проведено практичне моделювання та верифікацію за допомогою розробленого прототипу, результати якого задовольняють очікувані вихідні табличні дані роботи моделі для генерації та оцінювання тестів, котрі описані в другому розділі.

Проведено порівняння та аналіз результатів обробки запиту на сервері прототипу, та на сервері аналогу від «Moodle»

ВИСНОВКИ

У кваліфікаційній роботі магістра поставлено метою дослідити та вдосконалити моделі та методи тестування при проходженні тренінгу.

Представлено розроблену модель генерації персоналізованих тестів для електронного навчального контенту, включаючи основні компоненти та інструменти. Побудова адаптивного онлайн-тренінгу здебільшого вимагає використання компонентів відкритої оцінки, тобто компонентів, які потребують втручання наставників. У цьому випадку дуже важливою є повна інтеграція між інструментом для побудови адаптивного навчального контенту та системою, через яку він надається слухачам.

У зв'язку з цим пропонується інтелектуальна система навчання, генерації та оцінювання електронних навчальних тестів з використанням цілочисельного програмування, запропоновано концептуальну модель, що включає інструменти для створення інтерактивних ресурсів та оцінки накопичених попередніх знань. Описана система спрямована на визначення ряду питань різного ступеню складності із попередньо визначеного набору питань, які складатимуть тест.

Модель призначена для реалізації на практиці поетапного процесу створення та надання персоналізованих знань. За допомогою розробки макетів екранів було створено концептуальну модель та візуальний прототип системи навчання для створення персоналізованого навчального контенту. Основні інструменти моделі ілюструються екранами, які в основному служать для створення аналізуючих компонентів, необхідних для побудови профілю компетенцій та налаштування взаємозв'язку між трьома основними етапами, через які згодом пройде учень - аналіз попередніх знань, перехід через оптимізований навчальний контент на основі продемонстрованих компетенцій етап підсумкової оцінки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сатунина А. Е. Электронное обучение: плюсы и минусы // Современные проблемы науки и образования : журнал. — 2016. — № 1. — С. 89—90.
2. Ellis, Ryann K. (2012), *Field Guide to Learning Management*, ASTD Learning Circuits, archived from the original on 24 August 2014, retrieved 5 July 2012
3. Davis, B., Carmean, C., & Wagner, E. (2009). "The Evolution of the LMS : From Management to Learning". *The ELearning Guild Research*. 24.
4. Raza SA, Qazi W, Khan KA, Salam J (April 2021). "Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic: An Expansion of the UTAUT Model". *Journal of Educational Computing Research*. 59 (2): 183–208. doi:10.1177/073563312096042. ISSN 0735-6331. PMC 7509242.)
5. Long, Phillip D. (2004). "Learning Management Systems (LMS)". *Encyclopedia of Distributed Learning*. Thousand Oaks: SAGE Publications, Inc. pp. 291–293. doi:10.4135/9781412950596.n99. ISBN 9780761924517.
6. Wang, Qiyun; Woo, Huay Lit; Quek, Choon Lang; Yang, Yuqin; Liu, Mei (9 June 2011). "Using the Facebook group as a learning management system: An exploratory study". *British Journal of Educational Technology*. 43 (3): 428–438. doi:10.1111/j.1467-8535.2011.01195.x. ISSN 0007-1013.)
7. Chaiprasurt, Chantorn; Esichaikul, Vatcharaporn (5 July 2013). "Enhancing motivation in online courses with mobile communication tool support: A comparative study". *The International Review of Research in Open and Distance Learning*. 14 (3): 377–401. doi:10.19173/irrodl.v14i3.1416. ISSN 1492-3831.)
8. Lin, Sandi (16 November 2015). "SaaS Learning Management System: Is your LMS Truly SaaS? - eLearning Industry". *eLearning Industry*. Retrieved 4 February 2017.)
9. M.M. Glybovets "The role of standards in e-learning", *Computer technologies*, issue 148, vol. 160, pp. 107– 114, 2011. (in Ukrainian)

10. Yu.V. Stepanenko, A.P. Voichenko, "Features of digital learning objects repository in developing e-learning courses on mathematical disciplines". Bulletin CSU them. Petro Mogyla, issue 35, vol. 22, 2004. [Online]. Available: <http://lib.chdu.edu.ua/pdf/naukpraci/computer/2004/35-22-25.pdf> (in Ukrainian.)
11. SCORM® Overview. [Online]. Available: <https://adlnet.gov/scorm> (in English))
12. Advancing Learning Through Innovative Science and Technology. [Online]. Available: <http://adlnet.gov/> (in English)
13. IMS Global Learning Consortium. Learning Tools Interoperability. [Online]. Available: <http://www.imsglobal.org/activity/learning-tools-interoperability> (in English).
14. IMS Global Learning Consortium. Learning Tools Interoperability. [Online]. Available: <http://www.imsglobal.org/toolsinteroperability2.cfm..>
15. O.A. Shcherbyna, "Learning Tools Interoperability – a new standard for integration of distance learning platforms", Information Technologies and Learning Tools, vol 47, No 3, pp. 167–177, 2015. (in Ukrainian)..
16. What is the Experience API?. [Online]. Available: <https://xapi.com/overview/> (in English).
17. A. Bakharia et al. "Recipe for success: lessons learnt from using xAPI within the connected learning analytics toolkit", Proceedings of the sixth international conference on learning analytics & knowledge. ACM, pp. 378- 382, 2016. (in English).
18. O.A. Shcherbyna, "Experience API – new standard of e-learning software and examples of its practical use", Information Technologies and Learning Tools, vol. 53, No 3., pp. 150–163, 2016. (in Ukrainian).
19. "SCORM Content Aggregation Model, Version 1.2," Advanced Distributed Learning, 2002
20. "SCORM Content Aggregation Model, Version 1.3," Advanced Distributed Learning, 2004
21. Hyunah Kim, Eunjung Lee, Junchul Chun and Kwanghoon Pio Kim "A SCORM-based e-Learning Process Control Model and Its Modeling System" 2012

[Online] <https://www.researchgate.net/publication/220595189> A SCORM-based e-Learning Process Control Model and Its Modeling System

22. ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability

23. Дж. Пітерсон, Теорія мереж Петрі і моделювання систем, М.: Мир, 1984.