

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики  
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота  
на здобуття освітнього рівня бакалавра**

за спеціальністю 121 Інженерія програмного забезпечення

на тему:

**РОЗРОБКА СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ З  
ВИКОРИСТАННЯМ ФРЕЙМВОРКІВ SPRING ТА ANGULAR**

Виконала студентка 4-го курсу  
Олександра КМЕТЬ

\_\_\_\_\_  
(підпис)

Науковий керівник:  
доцент, кандидат фіз.-мат. наук  
Лариса КАТЕРИНИЧ

\_\_\_\_\_  
(підпис)

Засвідчую, що в цій роботі немає запозичень  
з праць інших авторів без відповідних  
посилань.

Студент

\_\_\_\_\_  
(підпис)

Роботу розглянуто й допущено до захисту  
на засіданні кафедри інтелектуальних  
програмних систем

« \_\_\_\_ » \_\_\_\_\_ 2021 р.,

протокол № \_\_\_\_

Завідувач кафедри

Олександр ПРОВОТАР

\_\_\_\_\_  
(підпис)

Київ – 2021

## РЕФЕРАТ

Обсяг роботи – 44 сторінки, 8 рисунків, 7 таблиць

АРХІТЕКТУРА «КЛІЄНТ-СЕРВЕР», ВЕБ-СЕРВІС, СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ, ANGULAR, LMS, MVC, JAVA, SPRING FRAMEWORK, POSTGRESQL

Об'єкт дослідження – організація та керування навчальним процесом в освітньому закладі.

Предмет роботи – програмна система, призначена для організації навчального процесу в закладі вищої освіти.

Метою роботи є автоматизація дистанційного виконання задач організації навчального процесу шляхом створення системи, призначеного для використання на технічній платформі закладу.

Інструменти розроблення: інтегроване середовище розробки IntelliJ IDEA, мови програмування Java, TypeScript, фреймворки Spring, Angular 9, СУБД PostgreSQL, Maven, Junit, Mockito.

Результат роботи:

- проведено аналіз предметної області та існуючих альтернатив
- спроектована архітектура проекту
- розроблено та реалізовано вимоги до системи.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ .....	5
ВСТУП .....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Дослідження предмету, цілей та особливостей веб-сервісів для дистанційного навчання .....	9
1.2 Аналіз існуючих аналогів.....	10
1.2.1 Сервіс Google Classroom .....	10
1.2.2 Системи керування навчанням (LMS) .....	12
РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ СИСТЕМИ..	16
2.1 Вибір моделі розробки.....	16
2.2 Вибір архітектури системи .....	17
2.3 Вибір мови програмування .....	19
2.3.1 Вибір мови програмування для серверної частини.....	19
2.3.2 Вибір мови програмування для клієнтської частини .....	21
2.4 Вибір СУБД .....	23
РОЗДІЛ 3 СПЕЦИФІКАЦІЯ ВИМОГ .....	27
3.1 Специфікація користувацьких вимог.....	27
3.2 Специфікація системних вимог .....	28
3.2.1 Функціональні вимоги.....	29
3.2.2 Вимоги до якості системи .....	36
3.3 Проектування бази даних .....	38
3.3.1 Опис таблиць бази даних .....	38
3.3.2 ER-модель .....	39

	4
3.4 Модульна декомпозиція .....	41
3.5 Моделювання управління .....	42
ВИСНОВКИ.....	43
ПЕРЕЛІК ПОСИЛАНЬ.....	44

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

БД – база даних

СУБД – система управління базою даних

МП – мова програмування

LMS – learning management system

MVC – Model View Controller

ОС – операційна система

ПЗ – програмне забезпечення

## ВСТУП

З появою Інтернет словосполучення «дистанційна робота» стало звичним для будь-якої сфери науки, культури та бізнесу. Вже нікого не дивують можливості доступу до електронних наукових бібліотек, онлайн консультації фахівців на спеціалізованих сайтах, включення експертів в телевізійні ефіри, перемовини політиків із застосуванням сучасних засобів дистанційної комунікації тощо. Сказане повною мірою стосується й дистанційного навчання в усіх галузях освіти.

Відповідно до Положень про дистанційне навчання [1] «дистанційне навчання — це індивідуалізований процес набуття знань, умінь, навичок і способів пізнавальної діяльності людини, який відбувається в основному за опосередкованої взаємодії віддалених один від одного учасників навчального процесу у спеціалізованому середовищі, яке функціонує на базі сучасних психолого-педагогічних та інформаційно-комунікаційних технологій».

Актуальність проблеми дослідження. Проблеми дистанційної роботи загалом, та дистанційного навчання зокрема, суттєво загострилися в зв'язку з пандемією SARS-COV-2, яка спричинила перенесення в режим онлайн процесів комунікації, які до цього часу відбувалися «в очному порядку». Це концерти, конкурси, конференції фахівців, консультації медиків, і, навіть, засідання тристоронньої контактної групи з мирного врегулювання ситуації на сході України. Та, видається, найбільших проблем пандемія додала освітнім процесам, перенесення яких в онлайн стало критичним як для викладачів (учителів), так і для учнів (студентів). Хоча є чимало різноманітних засобів дистанційної комунікації (Інтернет платформ, програмних систем та веб-застосунків), небагато з них є україномовними, безплатно доступними, простими у використанні та такими, що належним чином захищають особистий інформаційний простір користувачів. Тому проблема створення корпоративних платформ або застосунків, які

підтримують окремі види роботи в освітньому процесі є наразі актуальною проблемою програмної інженерії.

Стан розробленості наукової проблеми у вітчизняній та зарубіжній науці. Світова наука та технологія в галузі дистанційної комунікації пройшла довгий шлях від появи радіо (на початку 1900х років), телебачення (на початку 1950х років) до появи Інтернет наприкінці 80х років минулого століття та перших браузерів у 1993 році. У 1998 році перші онлайн навчальні курси було запроваджено у трьох освітніх закладах США. У 2000 році в процес дистанційного навчання було залучено майже 8% студентів повідних ВУЗів світу, у 2013 – 30%, а станом на 2018 – понад 45%. Вирішення проблем організації дистанційної освіти опікуються й провідні вітчизняні фахівці в галузі освіти та науки.

Очевидно, що на теперішній час невпинно зростає не лише кількість учасників процесів дистанційної освіти, але й розмаїття Інтернет технологій, що їх підтримують на сучасних технічних засобах, як от інфраструктура як сервіс (IaaS), платформа к сервіс (PaaS) та програмний застосунок як сервіс (SaaS). Існує чимало форм онлайн комунікацій та відповідних інструментів, запропонованих до використання для організації дистанційного навчання, серед яких відеоконференції, форуми, чати, блоги, електронна пошта; множина сучасних популярних платформ, як от Moodle, Google Classroom, Classtime, сервіси відеоконференцій, як от Zoom, Microsoft Teams, Google Meet, Skype, а також велика кількість web-сервісів, які забезпечують вирішення окремих задач навчального процесу, в основному стосовно створення контенту та його візуалізації. Проте задачі власне організації дистанційного навчання, специфічні для кожного освітнього закладу, як от планування навчального процесу потребують для свого вирішення створення корпоративних сервісно-орієнтованих за стосунків.

Завдання дослідження. В роботі поставлено й вирішено такі завдання:

1. проаналізувати функції існуючого інструментарію організації дистанційного навчання та сформулювати функціональні вимоги до

програмної системи; визначити прийнятні нефункціональні вимоги щодо її якості, безпеки тощо;

2. проаналізувати технічні особливості доступних платформ розроблення web-додатків та виробити архітектурні та проектні рішення стосовно системи, призначеного до розроблення;
3. виконати проектування та розробку системи;

## РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Дослідження предмету, цілей та особливостей веб-сервісів для дистанційного навчання

Основною метою систем дистанційного навчання є вдосконалення процесу навчання в режимі онлайн. Вони забезпечують можливість створення навчальних матеріалів, реєстрацію та адміністрування курсів, надають системи оцінювання тощо.

Такі системи розміщені в інтернеті та широко використовуються в навчальних закладах або компаніях для вдосконалення процесів викладання, методології та моніторингу процесу навчання.

Перевагою використання таких систем є можливість підвищення ефективності за допомогою інструментів відстеження та звітування. Прогрес нових користувачів можна відстежувати, записи можна переглядати, а користувачі можуть одночасно проходити декілька курсів. Навчання можна здійснити за допомогою веб-тренінгу. Керівництво може отримати доступ до записів за потреби виявити, яка область потребує вдосконалення. Учні також бачать свій прогрес та можуть відслідковувати свою успішність.

Деякі системи дистанційного навчання дозволяють викладачам створювати індивідуальні тести для учнів. Платформи дозволяють різні типи запитань, такі як: одно/багаторядкова відповідь, відповідь із множинним вибором, порядкова відповідь, правда чи хиба та інші. Деякі системи також дозволяють керувати відвідуваністю та записи про те, чи відвідував учень навчальний заклад, прийшов із запізненням чи пропустив заняття та заходи.

## 1.2 Аналіз існуючих аналогів

Розглянемо та порівняємо системи та сервіси, що наразі є найбільш широко вживаними в сфері дистанційної освіти.

### 1.2.1 Сервіс Google Classroom

Кожному користувачу Google крім таких сервісів, як Google Drive та Gmail надається доступ до веб-сервісу Google Classroom.

Google Classroom - веб-сервіс, доступний для кожного, у кого створений обліковий запис Google, нарівні з такими сервісами, як Gmail, Google Drive, Google Calendar та інші. Всі ці сервіси є необхідними для зручної роботи з Google Classroom.

При користуванні онлайн-сервісом для дистанційного навчання вам знадобляться такі інструменти, як Google Docs, Google Drive, Gmail та Sheets and Slides. Платформа надає можливості організувати будь-який навчальний процес через інтернет, використовувати розділення на класи або групи з можливістю додавати до них студентів. При необхідності існує функціонал додавання певних навчальних матеріалів або файлів, студенти можуть відправляти виконане домашнє завдання, яке потім можна буде перевірити чи виставили бали.

Викладач, відкривши вкладку «Завдання», може створювати та призначати завдання для створених класів, завдання можуть бути різних видів. На вкладці під назвою "Люди" за допомогою email-адресів можна запросити нових учнів до вашого курсу або додати ще одно викладача. Вкладка «Оцінки» відображає інформацію про навчальні модулі та відповідні їм оцінки учнів, що полегшує моніторинг навчального процесу.

Сервіс інтегрований з Google Drive, тож на диску будуть збережені всі файли. Google Drive надає широкий спектр функцій, таких як перегляд вмісту папок створених користувачем курсів, керування їх вмістом або поширення

доступу іншим учасникам навчального процесу. Але разом з тим необхідно відмітити, що такий функціонал займає досить багато пам'яті на диску, до того ж створення папок відбувається повільно.

Створення тестів і завдань учням. Якщо завдання створюються у вигляді гугл-документів, то за бажанням система може створити для кожного учасника курсу його індивідуальну копію файлу з завданням. Термін на виконання завдання може бути обмеженим. Коли учень виконує завдання та повертає його викладачеві, для документа проставляється статус "Перегляд", що вказує викладачеві на готовність завдання до перевірки. Викладач або відразу оцінює роботу та виставляє відповідні бали або повертає завдання студенту на доопрацювання.

Контроль за виконанням завдань. Надається можливість відслідковування прогресу у виконанні як одного конкретного завдання, так і всіх завдань одночасно.

Комунікація у класі. Для підтримки зв'язку між викладачем та студентами існує система створення оголошень, які будуть доступні для перегляду всім учасникам курсу. За необхідності залишити відгук, можна створити коментар у файлі учня чи використати електронну пошту.

Також слід відмітити істотні мінуси системи:

- Інтегровані сервіси, такі як Google Docs, Sheets and Slides та інші, полегшують процес створення навчальних матеріалів та завдань для викладача, проте користувацький інтерфейс для студентів може бути не є інтуїтивно зрозумілим. Також для таких начальних закладів, як школи, значним обмеженням може стати обов'язкова наявність облікового запису Google, адже вікова політика Google не дозволяє створення акаунтів особам молодше років;
- На відміну від створення завдання та оголошень, створення тестів більше складний процес. Обмежений вибір типів питань та складності з автоматичною перевіркою, що неможлива для певних видів питань значно ускладнюють організацію навчального процесу

та своєчасного оцінювання. Тому значній частині викладачів доводиться використовувати сторонні сервіси для створення інтерактивних тестів, наприклад Learningapps;

- Багато претензій також до самої системи оцінювання. Результати учнів заносяться в таблицю з оцінками, де дані розділені за групами та датами, що не є достатнім для зручної роботи з сервісом. Відсутня можливість вести журнали успішності для кожної групи та функціоналу для моніторингу поведінки студентів, все це ускладнює організацію навчального процесу;
- Масштабованість системи також виникає питання. Не зважаючи на той, факт що Google Classroom підтримує одночасне навчання великої кількості людей, після відмітки в 100 учнів зручність керування процесом через сторінку адміністратора стрімко погіршується. На відміну від деяких LMS систем, проблема не може бути вирішена встановленням додаткових плагінів;
- Неможлива комунікація в режимі реального часу, тому користувачам доводиться використовувати сторонні сервіси, такі як Viber, Skype та Telegram;
- Робота системи: Після початку курсів група учнів залишається сталою, що унеможлиблює додавання нових осіб в процесі навчання. Відсутня можливість розбиття груп на підгрупи в межах конкретного курсу. Система не витримує навантаження при одночасному надсиланні завдань великою кількістю студентів. Така сама проблема існує і при оцінюванні, одночасне надсилання оцінок може працювати зі збоями.

### **1.2.2 Системи керування навчанням (LMS)**

Необхідність в інструментах для віддаленого навчання призвела до створення сучасних LMS систем, що стали невід'ємною частиною реального

світу. Вони надають широкий спектр можливостей, що дозволяє вивести дистанційне навчання на якісно новий рівень.

Система управління навчанням (LMS) - це онлайн-система або програмне забезпечення, яке слугує для планування, виконання та оцінки конкретного навчального процесу. Інакше кажучи, це програмне забезпечення, що використовується в програмах електронного навчання і допомагає в адмініструванні, веденні документації, та моніторингу навчального процесу.

Системи управління навчанням використовуються для підтримки онлайн-співпраці через Інтернет. Коледжі та компанії використовують їх для проведення онлайн-тренінгів; корпорації використовують їх для навчальних цілей, а також для ведення обліку працівників.

Комунікація та зворотній зв'язок між викладачем та студентами. Вона побудована на кшталт соціальної мережі, що містить вбудовані чати або наприклад віртуальні кімнати для спілкування. Прикладом системи з реалізованим спілкуванням в режимі реального часу є система Source LMS, що пропонує власний модуль чатів, який в повній мірі відтворює функціонал сучасних месенджерів. В системі Moodle є можливість встановити окремий плагін з можливостями чатів, який повністю інтегрується з основними модулями програми. LMS система D2L Brightspace за допомогою панелі адміністратора курсу надає можливість спілкування в приватних чатах. Чати сучасних LMS систем реалізують функціонал для обміну медіа файлів чи документів.

Для організації навчання у молодших класах гарним вибором буде використання Seesaw, де розробники зосередились на комфорті спілкування вчителя й учнів та поточної перевірки завдань, тобто можна сказати, що це не зовсім повноцінна LMS. Цікавим розширенням, що активно заохочує дітей до спілкування, є використання камери, яке робить процес навчання більш інтерактивним та захоплюючим.

Система навчання. Системи надають функціонал для всього, що пов'язано зі створенням курсів, проведенням занять та перевірки результатів навчання. В залежності від обраної системи, можуть бути творенні модулі тестування чи керування розкладом.

Особливістю LMS систем є існування функціонального блоку, що відповідає за створення та керування курсами, що відрізняється від способу організації даних в сервісах на кшталт Google Classroom чи Microsoft Office 365, де навчальні матеріали являють собою окремі файли на диску. Хоча наприклад в системі Moodle відсутній конструктор контенту за замовчуванням, проте є можливість додатково встановити плагіни з потрібним функціоналом, таким як от окремий редактор курсів. Відсутня необхідність використання зовнішніх сервісів для зручного тестування, адже програмні модулі LMS систем надають усі можливості.

Системи дистанційного навчання підтримують необхідні атрибути очного навчання, такі як електронний журнал та видачу сертифікатів після проходження певних курсів.

Таблиця 1.1 порівнює можливості систем Source LMS, Moodle та D2L:

	<b>Moodle</b>	<b>D2L</b>	<b>SourceLMS</b>
Курси	так	так	так
Розділ на студентів та викладачів	так	так	так
Можливість створювати лекції, завдання	можна	можна	можна
SMS у системі	ні	ні	так
Підтримка групових чатів	окремий плагін	тільки приватні повідомлення	так

Різний доступ до завдань/тестів	ні	ні	так
Адмін-панель	так	ні	так
Створення тестів	так	так	так

Таблиця 1.1 – Порівняння сучасних систем керування навчанням

Враховуючи усе вищесказане, можна дійти висновку, що LMS-системи надають значно більші функціональні можливості, ніж такі сервіси, як Google Classroom. Слід зауважити, що Google Classroom є безкоштовним програмним забезпеченням, для використання якого потрібен лише обліковий запис Google, що використовується для створення класів та організації навчального процесу. Тим часом, готові системи бувають безкоштовними (наприклад Moodle LMS) або пропонують різні види оплати.

## РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ СИСТЕМИ

### 2.1 Вибір моделі розробки

Модель розробки - це схема виконання робіт і завдань на процесах, що забезпечують розробку, експлуатацію та супровід програмного продукту, і відображає життя програмного забезпечення, починаючи від формулювання вимог до припинення його використання.

Основні моделі розробки ПЗ:

- «Waterfall Model» - каскадний модель, або «водоспад»;
- «Code and fix» - модель кодування і усунення помилок;
- «Iterative Model» - ітеративна (або ітераційна) модель;
- «Incremental Model» - інкрементна модель;
- «V-model» - V-образна модель, розробка через тестування;
- «Prototype Model» - прототипна модель.
- «Spiral Model» - спіральна модель;
- «Chaos model» - модель хаосу;

Серед цих моделей найбільшою популярністю користуються п'ять основних, а саме: інкрементна, каскадна, ітеративна, V-образна та спіральна.

Оскільки розробка відбувається індивідуально, доцільним буде обрати каскадну модель, так як відсутність замовника дозволяє нівелювати недоліки цієї моделі.

Каскадна модель почала використовуватися одною з перших, ще з 1970х років. Вона являє собою чітке розбиття проектів на послідовні етапи та робить припущення, що при переході на наступний етап попередній вже повністю завершений та більше не потребує доопрацювання. Самі етапи зображені на рисунку 2.1.

До переваг моделі слід віднести легку контрольованість розробки та чітку фіксацію послідовних процесів.

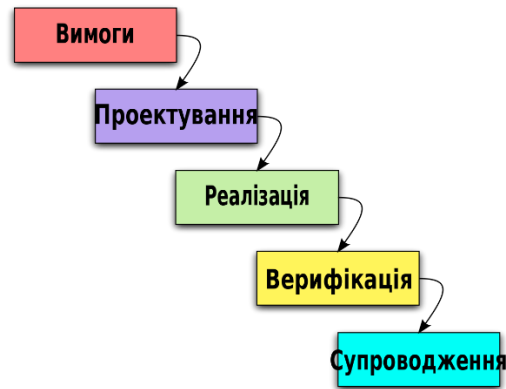


Рисунок. 2.1 - Каскадна модель

Недоліком є той факт, що замовнику видається готовий продукт, що не передбачає внесення додаткових змін. Проте, як було вказано вище, в даному випадку розробник також виступає в ролі замовника, тобто може вносити додаткові зміни на кожному етапі.

## 2.2 Вибір архітектури системи

Веб-сервіс – це тип програмного засобу, доступ до якого здійснюється за допомогою браузера користувача. Найбільш вживаним архітектурним шаблоном для програмного забезпечення даного типу є архітектура «клієнт-сервер».

Архітектура «клієнт-сервер» - устрій архітектури обчислювальних і мережних систем, при якому мережеве навантаження разом із завданнями розподілені між серверами, які виконують роль постачальників послуг, і клієнтами, що виконують роль сторони, яка запитує послуги.

Даний шаблон включає наступні компоненти:

- множину клієнтів, що запитують інформацію від серверів
- множину серверів, що займається обчисленням та наданням інформації
- мережу, необхідну для комунікації клієнтів та серверів

Варто відмітити, що і клієнти, як і сервери, є незалежними один від одного та можуть працювати паралельно. Також немає наперед визначених пар клієнт – сервер, клієнт може послідовно звертатися до будь-якого із серверів в довільній послідовності. Сервер в свою чергу може одночасно працювати із запитами з декількох клієнтів.

Класична архітектура клієнт-сервер включає два рівні - рівень клієнта та рівень сервера. Інший типовий дизайн клієнт-серверних систем використовує три рівні, які зображені на рисунку 2.2:

- Клієнт, який взаємодіє з користувачем
- Програмний рівень, що відповідає за бізнес-логіку програми
- Менеджер ресурсів, що використовується для зберігання даних

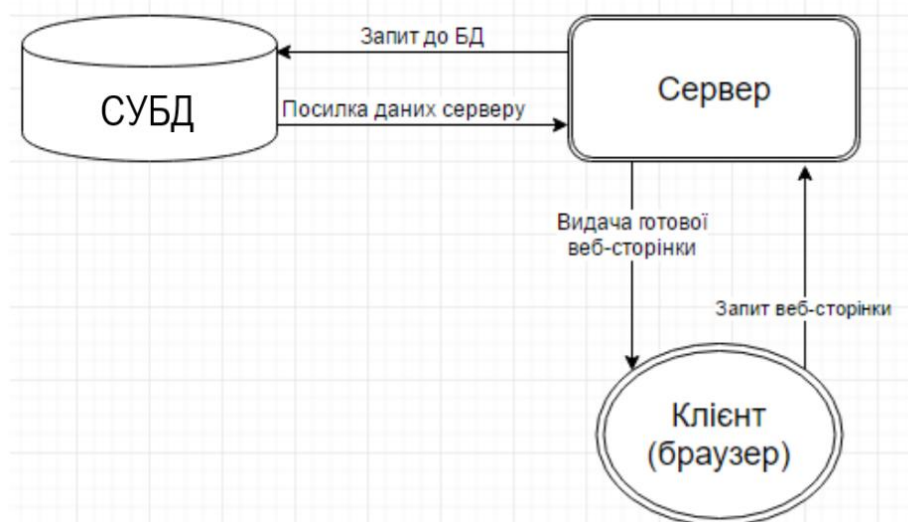


Рисунок 2.2 – Архітектура «клієнт – сервер»

Характерною рисою таких систем є відсутність необхідності завантаження та встановлення програм. В переважній більшості випадків можна використовувати вбудоване програмне забезпечення, а саме браузер. Інтерфейс з яким взаємодіє користувач і є клієнтом, що забезпечує доступ до системи та формує необхідні запити. Такий інтерфейс створюється у вигляді веб-сторінок.

Сервер здійснює обробку запитів, що надійшли з боку клієнта, та при необхідності звертається до сервера даних (менеджера ресурсів). Сервер даних здійснює базові операції з даними, такі як збереження, зміна,

видалення та вибірка згідно з запитом. Результати даних операцій передаються назад на програмний рівень. Дані, якими оперує менеджер ресурсів зазвичай зберігаються у вигляді реляційної бази даних.

Всі компоненти архітектури можуть розміщуватись як на одному, так і на декількох комп'ютерах.

## **2.3 Вибір мови програмування**

Більша частина широкоживаних мов програмування наразі дозволяють створювати та підтримувати веб-сервіси незалежно від їх рівня складності. Від вибору МП залежать наступні показники веб-системи:

- швидкість роботи
- швидкість та складність розробки
- необхідне устаткування для роботи

Не слід також забувати про досвід використання та рівень знань розробника в тій чи іншій мові програмування.

Оскільки обрана архітектура типу «клієнт-сервер», то необхідно створити два додатки, для кожного з яких має бути використана відповідна МП.

Результат виконання клієнтських мов програмування значною мірою залежить від браузера, адже виконується на стороні клієнта.

Серверні МП виконуються інакше: при відкритті користувачем сторінки (наприклад, використовуючи посилання) надсилається відповідний запит до сервера, де відбувається її обробка та виконання дій, передбачених бізнес-логікою, і в результаті сторінка знову вертається до клієнта по мережі.

### **2.3.1 Вибір мови програмування для серверної частини**

З огляду на навчальний план спеціальності та попередній досвід була обрана одна з найрозповсюдженіших мов програмування, а саме Java, яка

може застосовуватися для написання багатьох видів програм. Java – сильно типізована об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems [8]. Це гарний інструмент для побудови веб-застосунків адже має достатньо великий набір бібліотек та фреймворків, що полегшують процес розробки. Перевагою цієї мови також є кросплатформеність, що досягається за допомогою віртуальної машини.

Не менш важливою є надійність мови, що досягається завдяки жорсткій типізації та системі винятків і обробки помилок. Розробник може не втручатися в процес керування пам'яттю, оскільки вбудований збирач сміття займається цим самостійно.

В якості бібліотеки було обрано фреймворк Spring, який вважається одним з найсучасніших інструментів серед бібліотек Java призначених для розробки веб-додатків.

Spring – це написаний на мові Java універсальний фреймворк з відкритим кодом, що широко використовується для створення масштабних комерційних проектів. Він об'єднує в союі велику кількість додаткових модулів, серед яких планується використати наступні:

- Spring Boot

Spring Boot – це надбудова на фреймворком Spring, що дозволяє значно спростити процес конфігурації та розгортання додатків. Широко використовується в розробці веб-застосунків, оскільки вимагає мінімальних зусиль для початкових налаштувань та підтримує інтеграцію з іншими продуктами екосистеми Spring

- Spring MVC

Spring MVC – веб-фреймворк Spring, що надає можливість створення веб-сайтів та RESTful сервісів. Основною метою є забезпечення підтримки архітектури партерну «Model - View - Controller» для відокремлення об'єктів, бізнес-логіки та зовнішнього інтерфейсу один від одного.

- Spring JDBC

Spring JDBC – фреймворк з екосистеми Spring, що надає можливість реалізувати будь-яку взаємодію з базою даних. Більша частина низькорівневих операцій (наприклад, відкриття/закриття з’єднання) виконується автоматично, що значно зменшує громіздкість коду та полегшує читабельність.

- Spring Security

Spring Security – потужна система аутентифікації та контролю доступу, що легко конфігурується. Фактично вона є стандартом для захисту додатків створених на основі Spring. Основна мета фреймворку полягає в забезпеченні автентифікації та авторизації програм Java.

Поєднання продуктів, що входять в екосистему Spring, дозволяє значно прискорити розробку додатків та позбутися надлишкового коду [9].

### **2.3.2 Вибір мови програмування для клієнтської частини**

На сьогоднішній день серед веб-розробників найбільш популярною є мова JavaScript, яка вже стала стандартом для створення веб-застосунків. Тому саме цю МП було обрано в якості мови програмування для клієнтської частини. Також слід обрати фреймворк, що дозволить максимально ефективно використати мову.

Фреймворком називають програмне забезпечення, що полегшує розробку надаючи основу на якій можна створити власну програму. В залежності від вибору відповідного фреймворку може змінюватись сам процес розробки та складність підтримки готового продукту. Найкращим вибором буде надійний та сучасний фреймворк, що дозволяє розробляти нові рішення, адже сфера веб-розробки знаходиться в стані стрімкого розвитку.

Як показано на рисунку 2.3 найбільш популярними фреймворками серед веб розробників за версією GoogleTrends є React та Angular, тому будемо обирати серед них.

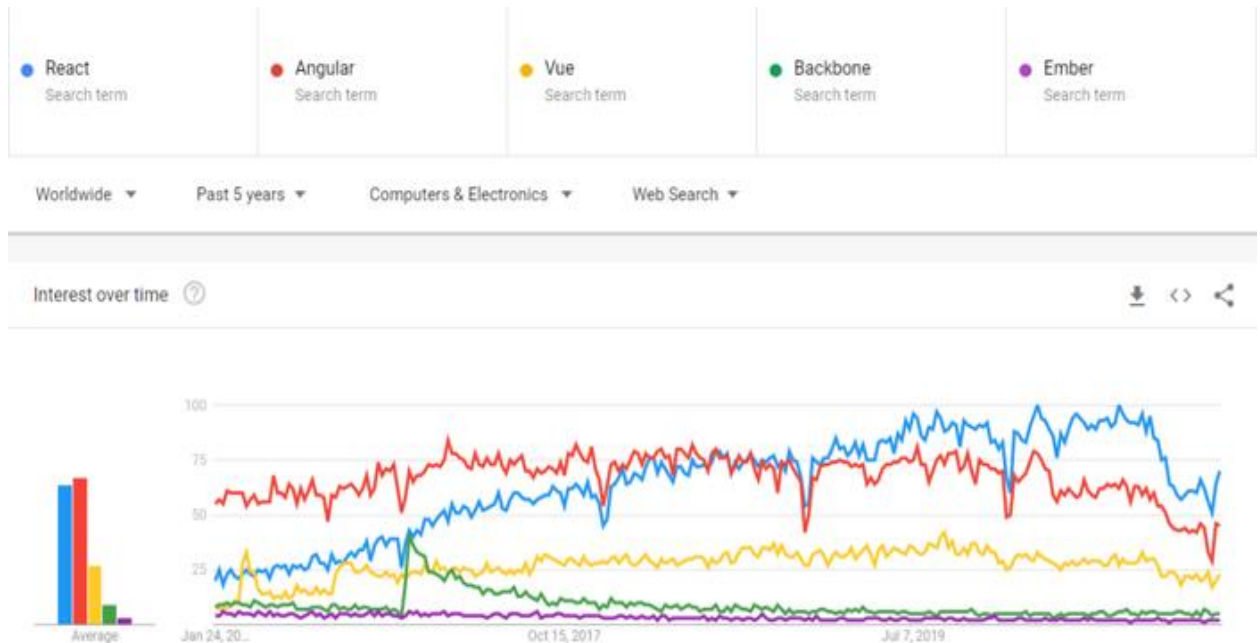


Рисунок 2.3 – Рейтинг веб-фреймворків

Обидва фреймворки забезпечують базові компоненти для веб-розробки. Також вони підпадають під ліцензію MIT і, отже, можуть бути використані безкоштовно.

І той і другий засновані на компонентах та надають можливість розбити додаток на менші компоненти використовуючи API. Не менш важливим є те, що обидва мають велику спільноту, яка підтримує фреймворки. Розглянемо переваги та недоліки кожного з них.

Angular - це фреймворк JavaScript розроблений Google, реліз якого відбувся у вересні 2016 року. Велика частина проектів Google розроблена саме за допомогою Angular. Серед популярних сайтів створених за допомогою даного фреймворку є Gmail, PayPal, Weather.com та Forbes [7].

Переваги:

- Дозволяє архітектуру MVC
- Створені програми мають високу продуктивність
- Нова концепція двостороннього зв'язування даних дозволяє мінімізувати кількість коду

- Надає інструмент командного рядка Angular-CLI, що пришвидшує створення та розширення проєктів
- Просте юніт-тестування

Недоліки:

- Повільний час завантаження програм
- Складність навчання
- Негнучка структура

React – це бібліотека JavaScript, розроблена Facebook у березні 2013го року. Він набагато популярніший у Facebook, ніж Angular у Google. Такі відомі сайти як Instagram, Netflix та Dropbox створені за допомогою React.

Переваги:

- Швидке завантаження сторінок
- Розмітка та логіка в одному файлі
- Легке навчання
- Надає рішення для мобільних додатків

Недоліки:

- Неможливість реалізувати архітектуру MVC
- Обмежені можливості, необхідно використовувати разом з іншими бібліотеками

Розглянувши переваги та недоліки двох найбільш широкочесно вживаних фреймворків та враховуючи попередній досвід розробника було прийнято рішення використовувати Angular, який є самодостатнім та дозволяє використовувати архітектуру MVC.

## 2.4 Вибір СУБД

Невід'ємною частиною додатків в будь-якій області, чи то мобільна розробка, чи веб або десктоп, є використання баз даних, оскільки необхідність в зберіганні та відтворенні даних є у кожного з них.

База даних – це систематизоване сховище даних, що підтримує зберігання та маніпуляції з ними. Кожна з них відноситься до певного типу (розподілені, реляційні, документно-орієнтовані, графічні та інші). Доступ та робота з базами даних відбувається за допомогою систем управління базами даних (СУБД).

Вони поділяються на 2 класи:

- реляційні
- нереляційні

Реляційні СУБД використовують мову SQL та визначають взаємозв'язки (relations) у вигляді таблиць. Нереляційні, або так звані NoSQL СУБД, перевизначають базові операції над даними та оперують документами формату JSON, що дозволяє зберігати великий об'єм даних в одній сутності, замість використання декількох таблиць.

Дані сервісу для дистанційного навчання чітко структуровані, при чому структура стабільна та не схильна до частих змін, тому в якості СУБД кращим буде використання реляційної СУБД.

Для даного проекту необхідно обрати безкоштовну реляційну СУБД, що не потребує великих зусиль для розгортання.

Рейтинг наведений на рисунку 2.4 показує, що найбільш популярними РСУБД наразі є:

- Oracle
- MySql
- Microsoft SQL Server
- Postgres SQL

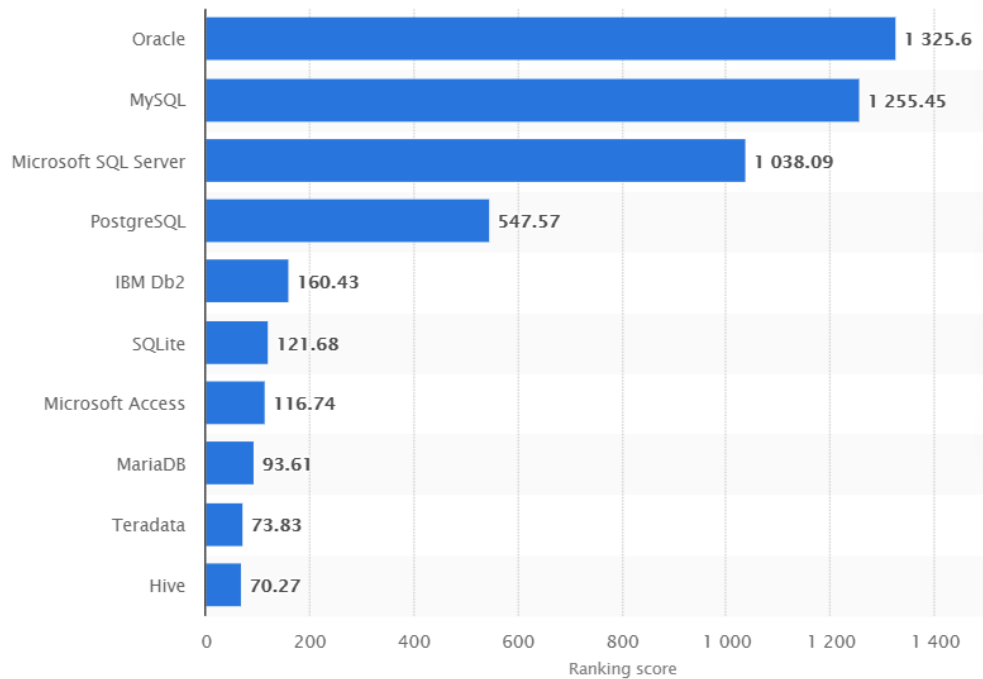


Рисунок 2.4 – Рейтинг реляційних СУБД 2020

Одразу можна відмітити, що Oracle не задовольняє умові про легку конфігурацію, адже для комфортної роботи з цією СУБД компаніям доводиться тримати у штаті адміністраторів Oracle. До того ж, об'єм необхідної пам'яті для якісного функціонування СУБД є завеликим для стандартних ПК. Тому дана СУБД не буде розглядатися в якості гарного вибору для проекту.

Також слід відмітити, що СУБД Microsoft SQL Server не є кросплатформеною та працює лише з операційною системою Windows, що є значним мінусом з огляду на те, що мова програмування для серверної частини була обрана саме за вищезазначену характеристику.

Тому необхідно зробити вибір між MySQL та Postgres SQL.

MySQL вважається однією з найпопулярніших СУБД з відкритою ліцензією. Найчастіше використовується для невеликих проектів, яким потрібен доступний засіб для роботи з БД. Підтримує різні типи таблиць і має велику кількість розширень.

PostgreSQL має дуже широкий функціонал. Буде гарним вибором для випадків, коли дані мають складну структуру або потрібно цілісність

збережених даних. Може застосовуватися як для невеликих, так і для громіздких комерційних проєктів.

Слід відмітити, що PostgreSQL - це об'єктно-реляційна база даних, тоді як MySQL - суто реляційна база даних. Це означає, що він підтримує такі функції, як наслідування та перевантаження, що може бути важливим для певних програм. PostgreSQL також більш дотримується стандартів SQL. Оптимізатор запитів та паралельність в PostgreSQL більш просунуті і набагато краще справляються зі своїми завданнями. Що стосується надійності та консистентності даних, то PostgreSQL переважає і цих параметрах.

Таким чином, враховуючи всі перераховані характеристики, в якості СУБД буде доцільним обрати PostgreSQL.

Підсумовуючи порівняння усіх вищенаведених технологій, можна дійти висновку, що система буде написана на мові Java з використанням фреймворку Spring, а саме таких його модулів, як Spring Boot, Spring MVC, Spring Data JDBC та Spring Security. Клієнтська частина додатку буде створена використовуючи фреймворк Angular і, відповідно, написана на мові TypeScript. В якості основної СУБД буде виступати PostgreSQL.

## РОЗДІЛ 3 СПЕЦИФІКАЦІЯ ВИМОГ

### 3.1 Специфікація користувацьких вимог

Короткий опис дійових осіб (акторів) надано в таблиці 3.1

<b>Актор</b>	<b>Короткий опис</b>
Неавторизований користувач	Може відвідувати лише головну сторінку, сторінки реєстрації та входу. Може зареєструватись в системі в якості викладача або студента чи увійти в систему. Може відновити втрачений пароль.
Авторизований користувач	Може редагувати свій профіль, змінювати та відновлювати пароль, спілкуватися в чатах групи
Викладач	Створює групи, створює та редагує в них оголошення, створює та переглядає тести.
Студент	Приєднується до групи, переглядає оголошення, проходить тести.

Таблиця 3.1 – Опис дійових осіб

Опис варіантів використання (далі – прецедентів) наведено в таблиці

### 3.2

<b>Код</b>	<b>Актор</b>	<b>Назва</b>	<b>Опис</b>
UU1	Неавторизований користувач	Реєстрація	Не авторизований користувач може зареєструватись в системі використовуючи свою пошту
UU2	Неавторизований користувач	Авторизація	Не авторизований користувач може авторизуватись використовуючи пошту та пароль
UU3	Неавторизований користувач	Відновлення паролю	При втраті користувачем паролю, він може його відновити використовуючи пошту

AU1	Авторизований користувач	Редагування свого профілю	Користувач може змінити своє прізвище, ім'я, опис.
AU2	Авторизований користувач	Зміна паролю	Користувач може змінити свій пароль.
AU3	Авторизований користувач	Спілкування в чаті	Користувач може надсилати і отримувати повідомлення в чаті групи в режимі реального часу
L1	Викладач	Створення групи	
L2	Викладач	Створення оголошень	Якщо викладач хоче чимось поділитись з студентами, він може створити публікацію
L3	Викладач	Створення тестів	Для перевірки знань студентів, викладач може створити тест
L4	Викладач	Перегляд тестів	Викладач може передивитись усі створені тести
S1	Студент	Приєднання до групи	Студент може приєднатись до групи за спеціальним кодом
S2	Студент	Перегляд оголошень	Студент може переглядати оголошення, розміщені викладачем в групі
S3	Студент	Проходження тестів	Студент може проходити тести, до яких його запросив викладач

Таблиця 3.2 – Опис варіантів використання

### 3.2 Специфікація системних вимог

Системні вимоги поділяються на функціональні та нефункціональні вимоги. Функціональні вимоги складаються з діаграми прецедентів (Use Case) та детального опису прецедентів, перелічених в таблиці 3.2.

Нефункціональними вимогами будемо називати вимоги до якості системи, такі як ефективність, безпека та інші.

### **3.2.1 Функціональні вимоги**

На рисунку 3.1 наведено діаграму прецедентів з можливостями, доступними різним видам користувачів в системі.

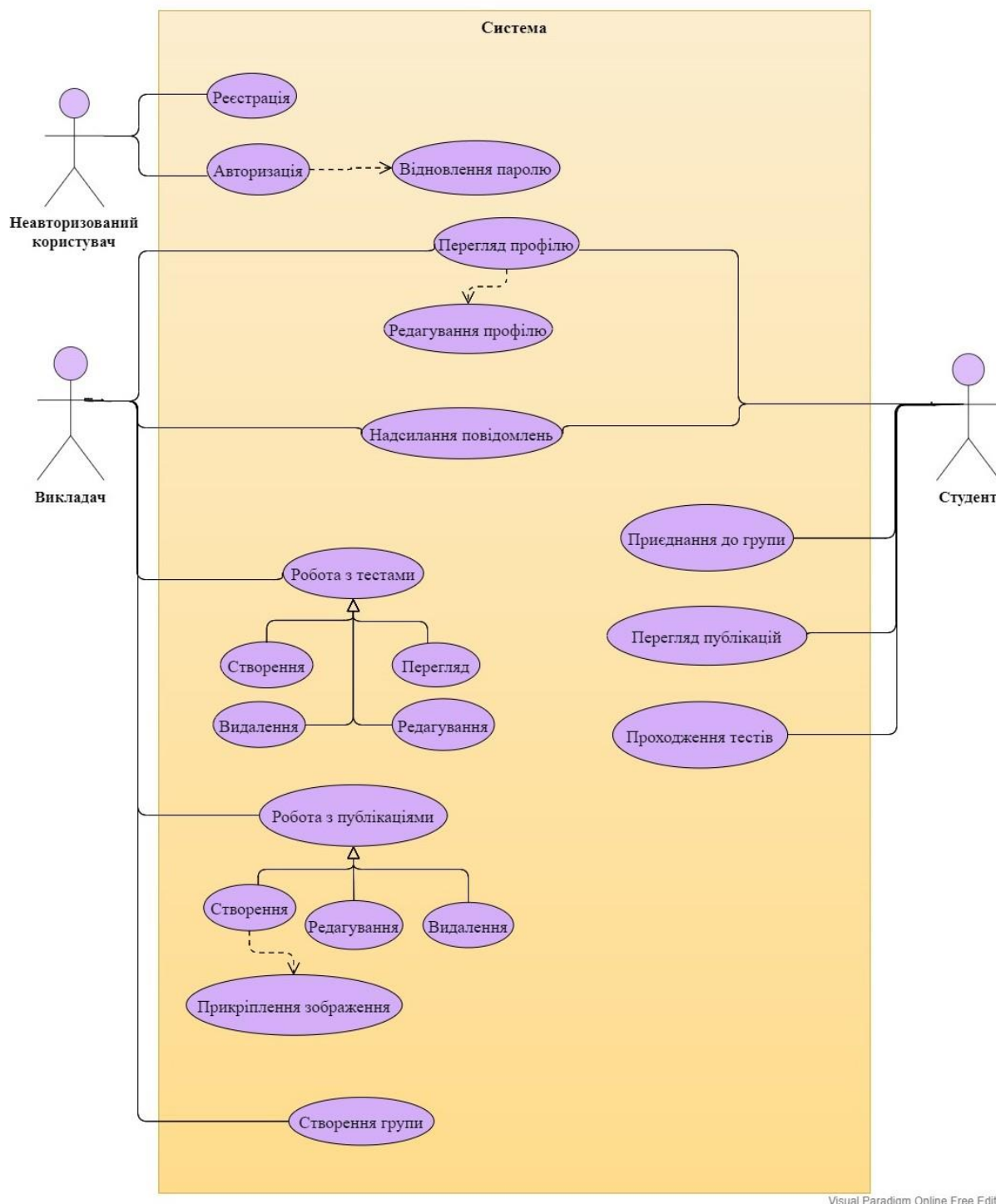


Рисунок 3.1 – Діаграма прецедентів

- Прецедент UU1: Реєстрація

На сторінці реєстрації є форма, в якій потрібно заповнити поля. Система надсилає на пошту посилання для підтвердження реєстрації. Коли користувач перейде за посиланням, система перевірить чи дійсний ще токен

і, якщо так, змінить статус юзера на «активований», користувач буде повідомлений про успішну реєстрацію.

Деталізація подій для даного прецеденту наведена на рисунку 3.2.

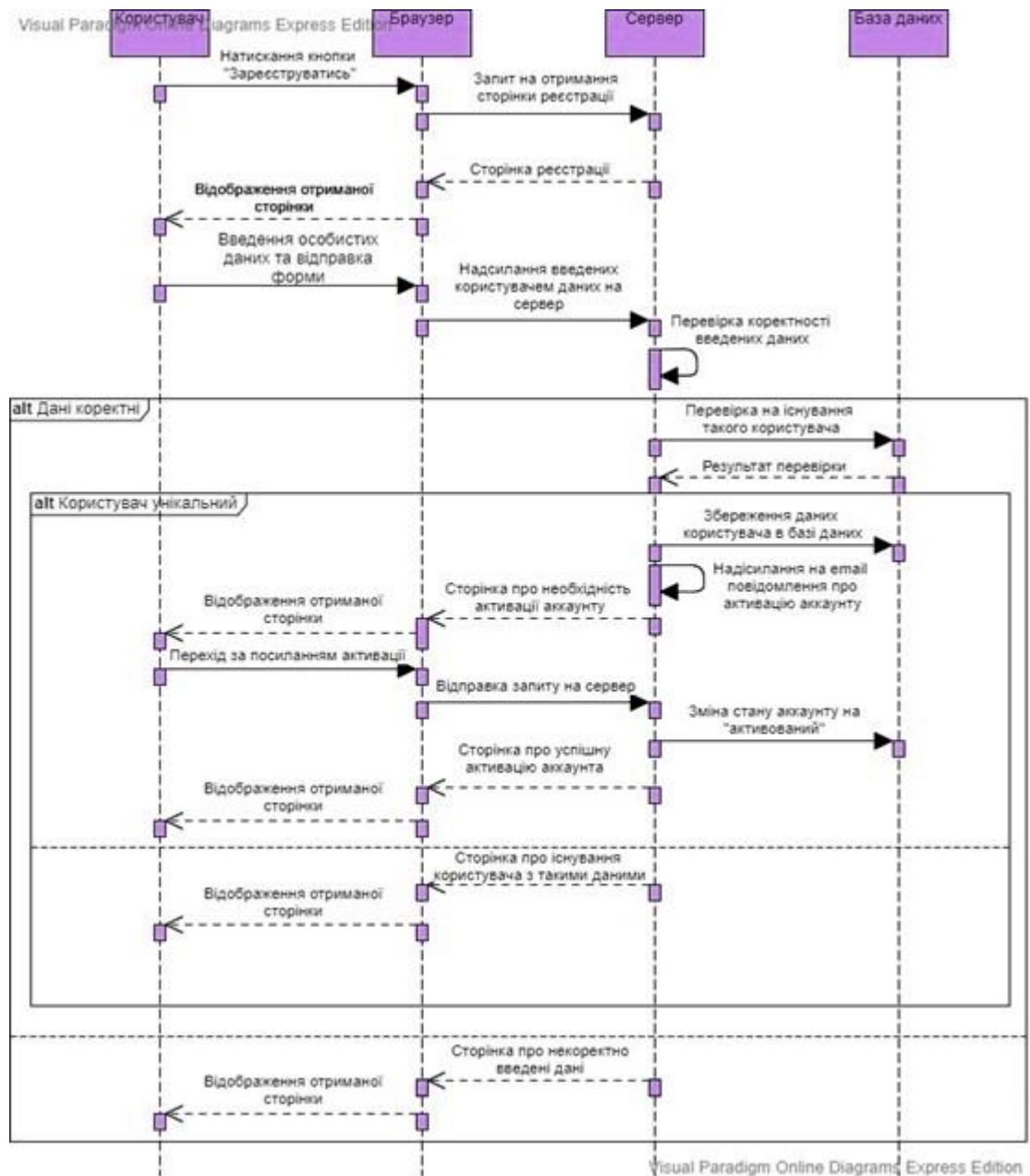


Рисунок 3.2 – Діаграма послідовностей UU1

Передумови – відсутні.

- Прецедент UU2: Вхід

Опишемо прецедент у вигляді потоку подій:

### 1.1.Вхід

#### 1.1.1. Короткий опис

На сторінці входу є форма, в якій потрібно ввести логін та пароль і натиснути кнопку відправки форми. Сервер здійснює аутентифікацію за введеними даними і у випадку успіху створює нову сесію для користувача. Також присутня можливість відновити пароль. Якщо користувач натисне «Забув пароль», на пошту буде надіслано посилання для введення нового паролю.

Діюча особа: Неавторизований користувач

## 1.2. Потік подій

Прецедент починається, коли Неавторизований користувач(далі НК) натискає “Вхід” в панелі меню

### 1.2.1. Базовий потік - Вхід

1. НК натискає “Вхід”
2. Система відображає сторінку з формою для вводу даних
3. НК вводить логін та пароль
4. НК натискає “Увійти”
5. Система перевіряє, чи існує введений логін в системі. В разі успішної перевірки переходить до п.6
6. Система шифрує введений пароль та звіряє зі збереженим в базі даних. В разі успішної перевірки переходить до п.7
7. Система перевіряє, чи активний профіль знайденого користувача. В разі успішної перевірки переходить до п.8
8. Система надсилає сторінку з профілем користувача

### 1.2.1 Альтернативні потоки

#### 1.2.1.1 Користувач з введеним логіном не знайдений

Якщо при виконанні п.5 система не знайшла користувача з введеним логіном в системі, то система надсилає сторінку з повідомленням “Логін чи пароль невірний”

#### 1.2.1.2 Пароль невірний

Якщо при виконанні п.6 пароль не збігається зі збереженим в базі даних, система надсилає сторінку з повідомленням “Логін чи пароль невірний”

#### 1.2.1.3 Профіль користувача не активований

Якщо при виконанні п.7 система статус профіля “неактивований”, то система надсилає сторінку з повідомленням “Профіль не активований, перейдіть за посиланням на пошті”

### 1.3 Спеціальні вимоги

Відсутні

### 1.4 Передумови

#### 1.4.1 Реєстрація

Перед початком прецеденту НК має зареєструватись та підтвердити пошту

- Прецедент UU3: Відновлення паролю

На сторінці авторизації є кнопка «Забув пароль», при натисканні користувачу відкривається нова сторінка з формою для заповнення даних про пошту, на яку було зареєстровано акаунт. Після заповнення форми сервер генерує та відправляє на пошту буде відправлення посилання для зміни паролю. При переході за отриманим посиланням відкривається сторінка з формою для введення нового паролю. Після підтвердження оновлений пароль зберігається в базі даних та буде доступна авторизація з новими реквізитами для входу.

Передумова - прецедент UU2

- Прецедент AU1: Редагування свого профілю

Після успішної авторизації користувач потрапляє на сторінку свого профілю. При натисканні кнопки «Редагувати» відкривається форма для редагування існуючої інформації. Після редагування користувач має натиснути кнопку «Зберегти» для відправки оновлених даних на сервер. Сервер оновлює інформацію базі даних та повертає користувачу сторінку профілю з новими даними.

Передумови – прецедент UU1, UU2

- Прецедент AU2: Зміна паролю

На сторінці свого профілю при натисканні кнопки «Редагувати» відкривається форма для редагування існуючої інформації про користувача. В поля «Новий пароль» та «Підтвердження нового паролю» користувач має ввести бажаний пароль. Після редагування користувач має натиснути кнопку «Зберегти» для відправки оновлених даних на сервер. Сервер оновлює інформацію бази даних та повертає користувачу сторінку профілю. Наступна авторизація має відбуватись з використанням нових реквізитів для входу.

Передумови – прецедент UU1, UU2

- Прецедент AU3: Спілкування в чаті

При відкритті сторінки чату клієнт під'єднується до веб-сокету та відправляє на сервер запит на отримання всіх повідомлень в відкритому чату. Користувач вводить повідомлення та натискає кнопку «Відправити». Клієнт по веб-сокету пересилає повідомлення на сервер, воно записується до бази даних та надсилається всім користувачам, приєднаним до веб-сокету.

Передумови – прецедент UU1, UU2 та L1 або S1

- Прецедент L1: Створення групи

На сторінці перегляду груп викладач може натиснути кнопку «Створити групу». Відкриється форма, в яку необхідно ввести назву групи. Після підтвердження інформація надсилається на сервер, де створюється новий об'єкт групи та унікальне посилання для приєднання до групи. Також сервер автоматично створює чат для групи та додає туди викладача. Вся інформація зберігається в базі даних, сервер повертає викладачу сторінку з переліком усіх створених ним груп.

Передумови – прецедент UU1, UU2

- Прецедент L2: Створення оголошень

Після натискання кнопки створити оголошення, відкривається форма з полями «Заголовок», «Підзаголовок» та «Зміст». Поля «заголовок» та «зміст» обов'язково мають бути заповнені. За бажанням користувач може прикріпити

фотографію, натиснувши кнопку «Додати зображення» та обравши файл з комп'ютера. Після підтвердження інформація надсилається на сервер, де створюється де зберігається в базі даних. Сервер повертає викладачу сторінку з переліком усіх створених ним оголошень в групі.

Передумови – прецедент UU1, UU2, L1

- Прецедент L3: Створення тестів

Сторінка створення тесту містить форму, де необхідно ввести назву тесту та є можливість додати теги з доступного переліку або створити новий. Для додавання питання до тесту потрібно натиснути кнопку «Додати питання» та обрати бажаний тип.

Питання можуть бути 4х типів:

- 1) З одним правильним варіантом відповіді
- 2) З декількома правильними варіантами відповіді
- 3) Завдання на встановлення послідовності
- 4) Введення відповіді

До кожного питання або варіанту відповіді можна прикріпити зображення. Після завершення потрібно натиснути кнопку «Створити тест». Клієнт перевіряє коректність введених даних та у разі позитивного результату перевірки відправляє дані на сервер, сервер зберігає тест в базу даних та повертає користувачу сторінку з усіма створеними їм тестами.

- Прецедент S1: Приєднання до групи

Студент вводить в форму спеціальний код для приєднання та натискає кнопку «Приєднатись до групи», введений код відправляється на сервер, де звіряється з існуючими в базі даних. В разі співпадіння, сервер записує користувача в базі даних як учасника групи та додає його до раніше створеного чату групи, де є викладач та інші учасники. Сервер повертає користувачу сторінку групи.

Передумови – прецедент UU1, UU2 та L1 (іншим користувачем)

- Прецедент S3: Проходження тестів

При приєднанні до проходження тесту, користувач потрапляє на сторінку очікування усіх учасників. Клієнт при цьому під'єднується до веб-сокету, та відправляє повідомлення про приєднання. Коли викладач починає тест, студент по веб-сокету отримує питання та час на відповідь. При завершенні часу відповідь автоматично відправляється на сервер, де звіряється з правильною. Після завершення тесту сервер підраховує загальні бали учасників та надсилає кожному сторінку з результатами.

### 3.2.2 Вимоги до якості системи

В таблиці 3.3 представлено опис користувацьких нефункціональних вимог.

<b>Вимога</b>	<b>Опис</b>
Точність	Час отримання та надсилання повідомлення повинен бути вказаний з точністю до хвилин.
Своєчасність	Час появи оголошення в групі після його створення та час отримання користувачем повідомлення, яке було надіслано іншим користувачем має не перевищувати 3 секунди з моменту створення/надсилання.
Зручність	Інтерфейс системи має бути зручним, інтуїтивно зрозумілим та не вимагати додаткової підготовки користувачів.

Таблиця 3.3 – Нефункціональні користувацькі вимоги

В таблиці 2 представлено опис некористувацьких нефункціональних вимог.

<b>Вимога</b>	<b>Опис</b>
Ефективність.	Сервіс не вимагає від користувача завантаження будь-якого програмного забезпечення на свій

	комп'ютер. Сервіс не вимагає від користувача доступу до геолокації та інших подібних сервісів.
Сумісність.	Соціальна мережа має бути сумісною із наступними браузерами: Google Chrome, Mozilla Firefox, Opera, Microsoft Edge, Yandex Browser
Переносимість.	Сервер соціальної мережі має без проблем запускатись на Unix-подібних операційних системах та Windows без зміни вихідного коду та з мінімальною зміною конфігурації.
Безпека. Повинні виконуватися наступні умови:	<p>Система не повинна розповсюджувати особисті дані користувача для загального огляду.</p> <p>Система не повинна могла випадково втратити інформацію про обліковий запис користувача.</p> <p>Система не повинна могла випадково втратити інформацію про публікації користувача.</p> <p>Система не повинна могла випадково втратити інформацію про діалоги користувача з іншими користувачами та їх вміст</p>
Системні вимоги до сервера	<p>Рекомендовані системні вимоги для запуску сервера соціальної мережі:</p> <p>Процесор: 4 ядерний</p> <p>Оперативна пам'ять: 8 Гб</p> <p>Вільне місце на жорсткому диску: 2 гб</p> <p>Встановлений jdk 11</p>

Таблиця 3.3 – Нефункціональні некористувацькі вимог

### 3.3 Проектування бази даних

Оскільки для проекту була обрана реляційна база даних, то основними її компонентами будуть таблиці, що слугують для структуризації та збереження інформації.

На основі описаних вище функціональних вимог була спроектована база даних, що реалізує модель «Сутність – Зв'язок» та створена використовуючи СУБД PostgreSQL.

#### 3.3.1 Опис таблиць бази даних

В процесі проектування було розроблено 17 таблиць, 13 з яких є сутностями та 4 в якості реалізації зв'язку «багато-до-багатьох».

Спроектовані сутності:

- користувач;
- група;
- оголошення;
- чат;
- повідомлення;
- сповіщення;
- тест;
- питання;
- варіант відповіді;
- токен;
- тег;
- зображення;

Кожній сутності відповідає первинний ключ, необхідний для унікальної ідентифікації екземпляра сутності.

Таблиці для реалізації зв'язку «багато-до-багатьох»:

- тест – тег
- учасники чату
- учасники групи
- питання – відповідь

Такі таблиці не мітять первинного ключа, проте посилаються на первинні ключі сутностей.

Спроектвана база даних виконує умови третьої нормальної форми, тобто є нормалізованою. Умова полягає в незалежності ключових стовбців від інших ключових стовбців та виконанні критеріїв першої та другої нормальної форми.

Перша нормальна форма полягає в атомарності атрибутів, тобто дані в стовбцях є неподільними. Друга нормальна форма потребує виконання першої та вимагає залежності кожного не ключового стовбця від цілого ключового поля.

### **3.3.2 ER-модель**

Діаграма взаємозв'язків сутності (ERD, entity relationship diagram) показує зв'язки наборів сутностей, що зберігаються в базі даних. Сутність у даному контексті є об'єктом, компонентом даних. Набір сутності - це сукупність подібних сутностей. Ці сутності можуть мати атрибути, що визначають його властивості.

Визначаючи сутності, їх атрибути та показуючи взаємозв'язки між ними, ER діаграма ілюструє логічну структуру баз даних.

На рисунку 3.3 наведена діаграма взаємозв'язків сутності, що ілюструє відношення таблиць і взаємодію бази даних спроектованого сервісу для дистанційного навчання,

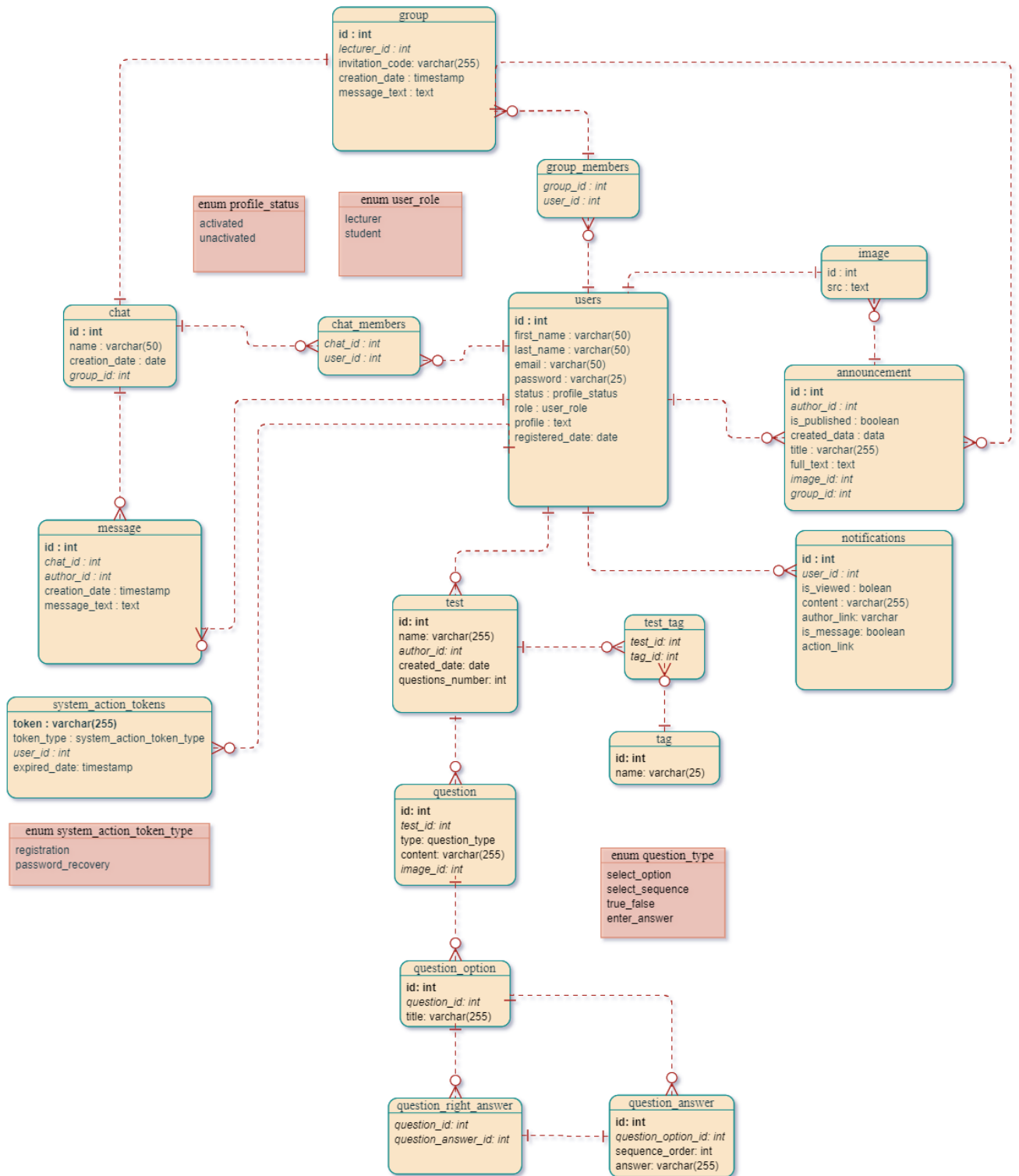


Рисунок 3.3 – ER діаграма

### 3.4 Модульна декомпозиція

В проєкті було вирішено використовувати фреймворк Spring, в основі якого лежить архітектура патерна Model-View-Controller, що реалізована за допомогою слабо зв'язаних готових компонентів. Патерн MVC розділяє модулі програми (дані, бізнес-логіку і логіку UI), забезпечуючи при цьому вільний зв'язок між ними.

Такими компонентами є:

- Model (Модель) інкапсулює (об'єднує) дані програми, загалом вони будуть складатися з POJO («Старих добрих Java-об'єктів», або бінів).
- View (Відображення) відповідає за відображення даних Моделі, як правило, генеруючи HTML, які ми бачимо в своєму браузері.
- Controller (Контролер) обробляє запит користувача, створює відповідну Модель і передає її для відображення у View.

Вміст модулів програми та зв'язок між ними проілюстровано на рисунку 3.4.

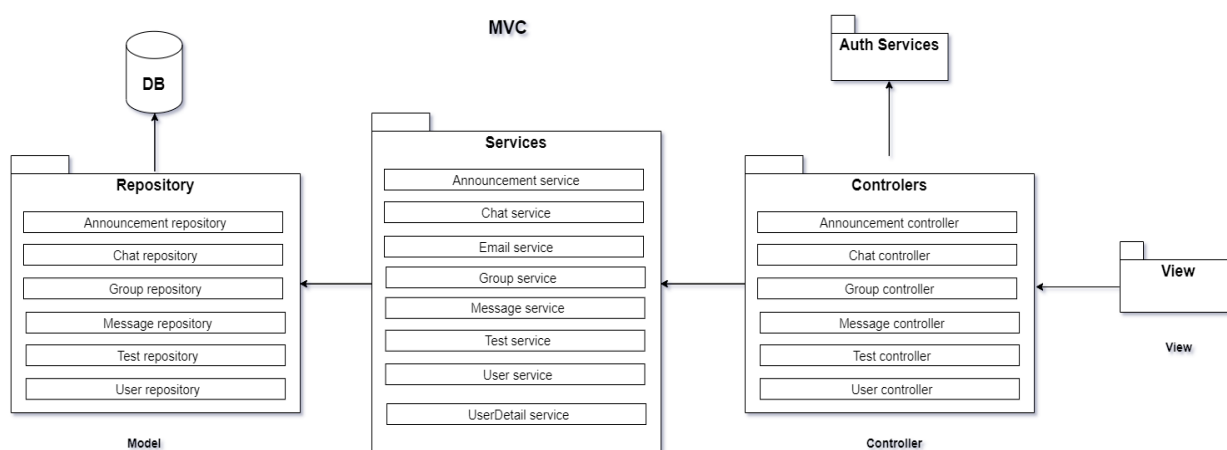


Рисунок 3.4 – Модулі програми

### 3.5 Моделювання управління

Для демонстрації процесу обробки запитів користувача різних видів використаємо діаграму на рисунку 3.5:

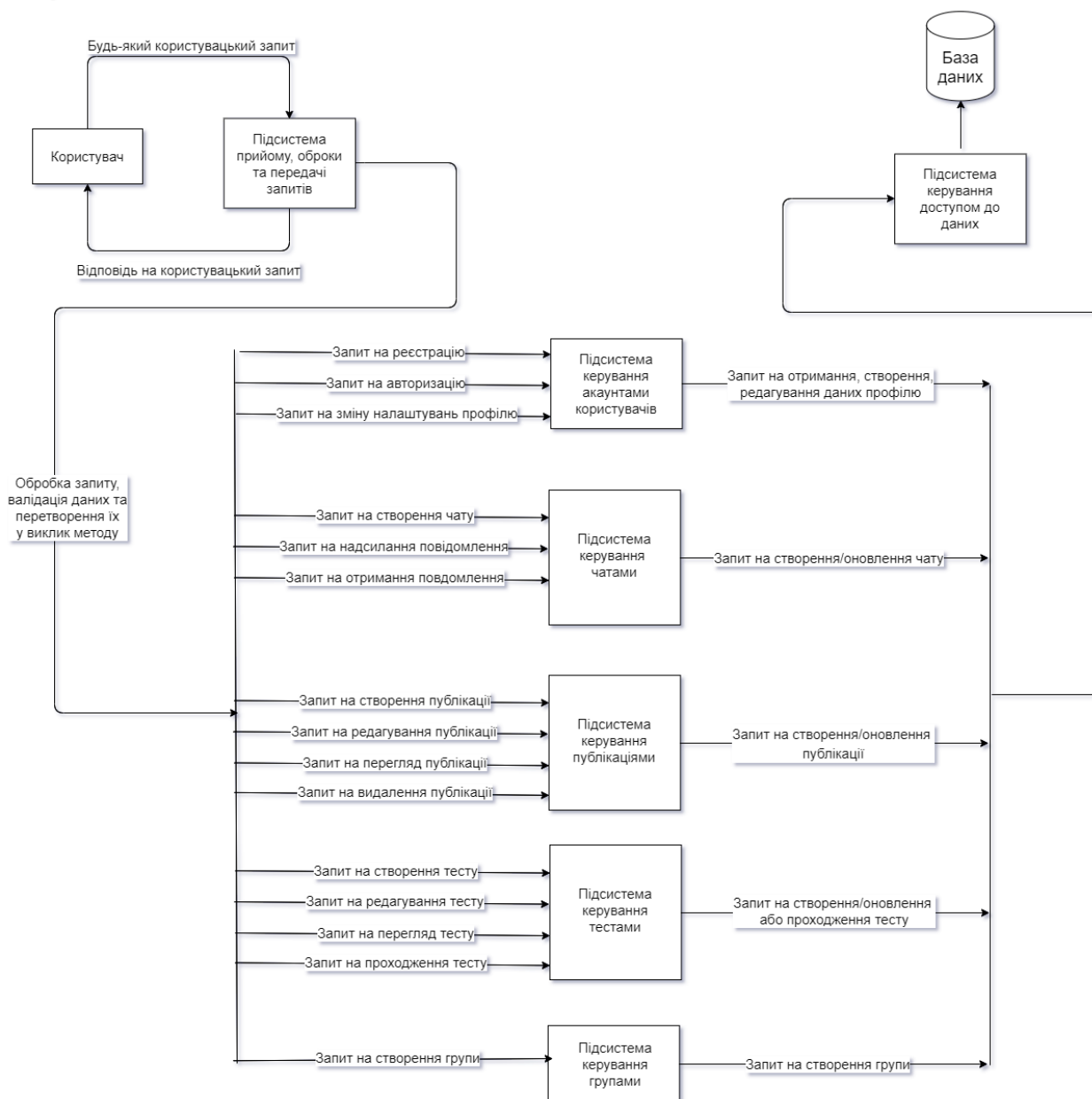


Рисунок 3.5 – діаграма управління

Таким чином, можна побачити, що саме відбувається на кожному етапі обробки запиту користувача від створення запиту клієнтів до моменту запису нової інформації в базу даних.

В розділі було сформульовано функціональні та нефункціональні вимоги та визначена структура системи.

## ВИСНОВКИ

В роботі було розглянуто проблеми дистанційного навчання та проаналізовано функції існуючого інструментарію організації навчання в режимі онлайн, на основі яких було сформульовано функціональні вимоги до програмної системи та визначено прийнятні нефункціональні вимоги щодо її якості, безпеки та інших параметрів.

Проаналізовано технічні особливості доступних сучасних платформ розроблення веб-додатків та обрано технології для реалізації системи. Вироблено та аргументовано архітектурні та проектні рішення стосовно системи, призначеного до розроблення.

В результаті кваліфікаційної роботи було автоматизовано дистанційне виконання задач організації навчального процесу шляхом створення системи, призначеного для використання на технічній платформі закладу.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Наказ Міністерства освіти і науки України від 25.04.2013 № 466 “Про затвердження Положення про дистанційне навчання”. Зареєстрований у Міністерстві юстиції України 30 квітня 2013 р. за № 703/23235 [Електронний ресурс] – Режим доступу до ресурсу: <https://mon.gov.ua/storage/app/media/zagalna%20serednya/metodichni%20recomendazii/2020/metodichni%20recomendazii-dustanciyna%20osvita-2020.pdf>
2. Стаття «A review of online education and services» [Електронний ресурс] – Режим доступу до ресурсу: <https://xplorexit.com/review-online-education-web-services-consideration-cloud-technologies-learning-systems/>
3. Стаття «Нова українська школа. Практики та підходи до дистанційного навчання.» [Електронний ресурс] – Режим доступу до ресурсу: <https://nus.org.ua/articles/praktyky-ta-pidhody-do-dystantsijnogo-navchannya-rekomendatsiyi-dlya-vchyteliv/>
4. Стаття «Засоби організації дистанційного навчання. Розділ 2. Засоби та інструментарій дистанційного навчання.» [Електронний ресурс] – Режим доступу до ресурсу: <https://uahistory.co/lesson/organization-remote-teaching-at-school-methodical-recommendations/3.php>
5. Онлайн інструменти для організації дистанційного навчання в умовах карантину. Інститут інформаційних технологій і засобів навчання НАПН України. Інформаційний бюлетень №2.-2020 [Електронний ресурс] – Режим доступу до ресурсу: [https://lib.iitta.gov.ua/719816/1/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9%20%D0%B1%D1%8E%D0%BB%D0%B5%D1%82%D0%B5%D0%BD%D1%8C%20%E2%84%962\\_2020.pdf](https://lib.iitta.gov.ua/719816/1/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B8%D0%B9%20%D0%B1%D1%8E%D0%BB%D0%B5%D1%82%D0%B5%D0%BD%D1%8C%20%E2%84%962_2020.pdf)
6. Java Web Services Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.javatpoint.com/java-web-services-tutorial>

7. Angular Docs [Электронный ресурс] – Режим доступа до ресурсу:  
<https://angular.io/docs>
8. Java Documentation [Электронный ресурс] – Режим доступа до ресурсу:  
<https://docs.oracle.com/en/java/>
9. Spring Framework Documentation [Электронный ресурс] – Режим доступа  
до ресурсу: [https://docs.spring.io/spring-  
framework/docs/current/reference/html/](https://docs.spring.io/spring-framework/docs/current/reference/html/)