

Міністерство освіти та науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

КУРСОВА РОБОТА
ПОЯСНЮВАЛЬНА ЗАПИСКА
з навчальної дисципліни “Аналіз та рефакторинг коду
програмного забезпечення”

Тема роботи: Програмна система для віддаленого навчання дітей в середній
школі (5-11 клас)

Студент. гр. ПЗП-17-7


(підпис)

Руденко Д.О.

Керівник роботи:

(підпис) доц. Лещинський В.О.

Роботу захищено «__»_____2020 р.
з оцінкою _____

Комісія:

(підпис) доц. Лещинський В.О.

(підпис) доц. Лещинська І.О.

(підпис) ст. викл. Сокорчук І.П.

Харків
2020 р.

Харківський національний університет радіоелектроніки
Факультет комп'ютерних наук Кафедра програмної інженерії
Спеціальність 6.050103 – Інженерія програмного забезпечення
Курс 3 Семестр 6
Навчальна дисципліна Аналіз та рефакторинг коду програмного забезпечення

ЗАВДАННЯ НА КУРСОВУ РОБОТУ СТУДЕНТОВІ

Руденку Дмитру Олександровичу

1. Тема роботи: «Програмна система для віддаленого навчання дітей в середній школі (5-11 клас)»
2. Термін узгодження завдання курсової роботи «__» _____ 2020 р.
3. Термін здачі студентом закінченої роботи «__» _____ 2020 р.
4. Вихідні дані по проекту (роботи): Онлайн-платформа, яка дозволяє створювати та редагувати навчальні курси та відповідні уроки до них, дає можливість об'єднувати учнів у групи та перевіряти відправлені роботи. Використовувати ОС macOS Catalina, NoSQL базу даних MongoDB, середовище розробки Visual Studio Code, середовища розгортання Heroku, Mongo Atlas, Netify, IOS/Android симулятор Expo
5. Зміст пояснювальної записки (перелік питань, що належить розробити) вступ, аналіз предметної галузі, постановка задачі, архітектура проекту, загальна архітектура системи, архітектура та проектування back-end серверу, проектування бази даних, архітектура мобільного пристрою, опис програмних рішень, опис розробленої програмної системи, висновки, перелік посилань, додатки
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) архітектура back-end серверу, схема бази даних, діаграма класів, інтерфейси створення користувача, інтерфейси навігації по курсам та урокам, інтерфейс мобільного додатку.

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів курсової роботи	Строк виконання етапів роботи	Примітки
1	Функціональна специфікація програмного проекту	20.02.2020	<i>виконано</i>
2	Проектування програмного проекту	27.02.2020	<i>виконано</i>
3	Кодування програмного проекту	05.05.2020	<i>виконано</i>
4	Оформлення пояснювальної записки	15.06.2020	<i>виконано</i>
5	Захист курсової роботи	28.06.2020	

Дата видачі завдання «__»_____ 2020 р.

Керівник _____ доц. Лещинський В.О.
(підпис)

Завдання прийняв до виконання
ст. гр. ПЗП-17-2  Руденко Д.О.
(підпис)

РЕФЕРАТ

Пояснювальна записка: 36 с., 8 рис., 1 дод., 7 джерел.

ОСВІТА, ВИКЛАДАЧ, УЧЕНЬ, КУРС, УРОК, REACT, NODE, WEB-САЙТ, MongoDB, NO-SQL, IOS, REACT NATIVE

Об'єктом дослідження є платформа для забезпечення зручного навчального процесу користувачів-учнів та гнучкого створення онлайн-уроків користувачами-викладачами.

Метою роботи є розробка системи, яка підтримує створення, редагування, видалення та читання курсів, уроків, а також дає змогу проводити оцінювання робіт та вести облік успішності учнів. У цю систему входить:

- а) серверна частина;
- б) веб-частина;
- в) мобільний додаток;

Методи розробки базуються на технології Node.JS [6] та React [5], сервер бази даних MongoDB [4], розміщений за допомогою хмарного провайдера AWS.

У результаті роботи здійснено програмну реалізацію системи яка підтримує створення, редагування, видалення та читання курсів, уроків, а також дає змогу проводити оцінювання робіт та вести облік успішності учнів.

ЗМІСТ

Вступ.....	6
1 Аналіз предметної галузі.....	7
1.1 Аналіз конкуренції.....	7
1.2 Бізнес потреби та пріоритетність	8
1.3 Середовище реалізації.....	9
2 Постановка задачі.....	10
2.1 Постановка мети.....	10
2.2 Основний функціонал системи.....	10
2.3 Допущення та залежності.....	11
2.4 Релізи.....	12
2.5 Користувацькі обмеження.....	13
3 Архітектура проекту.....	14
3.1 Загальна архітектура проекту.....	14
3.2 Архітектура та проектування back-end серверу.....	16
3.2 Архітектура та проектування бази даних.....	18
4 Опис програмних рішень.....	21
4.1 Back-end сервер.....	21
4.2 Front-end частина.....	22
4.3 Мобільний додаток.....	24
Висновки.....	26
Перелік джерел посилань.....	27
ДОДАТОК А Специфікація ПЗ.....	28

ВСТУП

У сучасному світі під впливом пандемії COVID-19 тенденції активно рухаються у напрямку дистанційних технологій та сервісів. Галузь освіти одна з найбільших, що потрапили під цей вплив.

Онлайн-платформи для навчання активно використовуються студентами в усьому світі, але серед школярів навчальної та середньої школи цей підхід поки що не ціниться у повній силі. Цей факт обумовлений тим, що більшість у цьому віку лише починають знайомитися з комп'ютером та взаємодією з ним.

Але завдяки сучасним технологіям існує можливість створити платформу, якою можна користуватися з мінімальним досвідом користування комп'ютерними пристроями, завдяки інтуїтивно зрозумілому інтерфейсу та яка буде мати максимальний рівень залучення уваги учнів завдяки гейміфікації та інтерактивності завдань.

Критерії успіху даної системи базуються на тому, що система надає можливість у разі необхідності повністю забезпечити дистанційне навчання учнів шкіл, а також забезпечує можливість проведення дистанційних занять з репетитором або персональним викладачем.

Метою роботи є реалізація робочої системи, яка дозволить викладачам створювати уроки, об'єднувати їх у курси, надавати доступ окремим учням або цілим групам учнів, а також проводити зручне оцінювання виконаних робіт; з іншого боку, учням буде дозволено переглядати доступні курси, завантажувати виконані роботи у систему, слідкувати за успішністю та контролювати терміни здачі завдань.

Сервіс повинен мати багаторівневу архітектуру, бути багат шаровим, масштабуватися, на клієнтському рівні мати веб-клієнт та мобільний додаток, мати можливість резервного копіювання даних та надавати захист персональної інформації користувачів та створеного ними контенту.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз конкуренції

Користувачами програмного продукту будуть люди, які викладають у школах та мають потребу у проведенні дистанційних занять, репетитори та персональні викладачі, які хочуть розширити базу учнів за межі місця проживання та проводити повноцінні онлайн-заняття.

Найбільшим конкурентом у галузі онлайн-освіти для школярів є продукт корпорації Google, що має назву Classroom [2]. Недоліком цієї платформи є те, що сама система є лише агрегацією існуючих сервісів Google, таких як: Drive, Calendar, Gmail. Тобто платформа лише забезпечує створення навчальних матеріалів у вигляді документів, спілкування викладача та учня та обліку термінів за допомогою календаря.

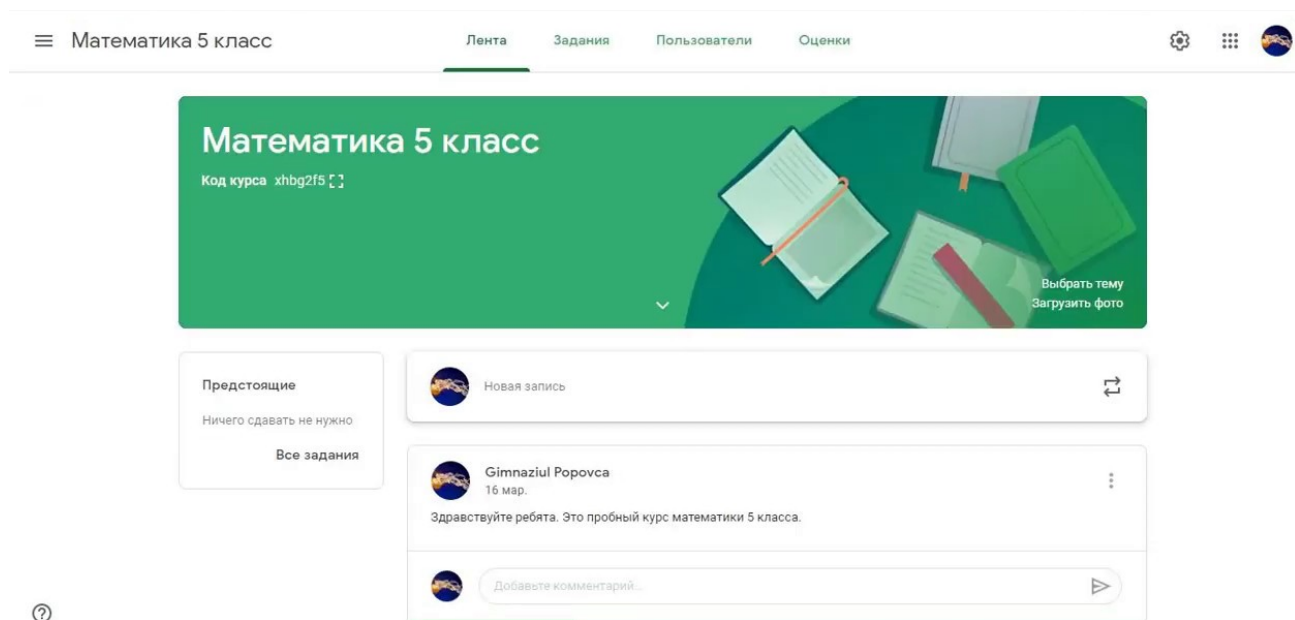


Рисунок 1.1 – Інтерфейс веб-додатку Google Classroom

Унікальність нашої платформи полягає у тому, що викладач матиме змогу створювати інтерактивні уроки та не обмежуватися стандартними документами або тестами. Також система матиме автоматичну систему оцінювання та ведення

статистики успішності учнів. Однією з ключових функцій є те, що викладачі можуть викладати свої уроки та курси у відкритий доступ для інших викладачів, що зменшить навантаження та труднощі викладачам-початківцям.

Під час розробки системи можуть виникнути наступні проблеми:

- а) відсутність учня користуватися комп'ютером та мережею Internet
- б) збій у роботі бази даних може призвести до втрати створеного користувачем контенту
- в) активне збільшення функціоналу існуючими онлайн-платформами у відповідній галузі

1.2 Бізнес-потреби та пріоритетність

Даний продукт може залучити користувачів своєю унікальністю та актуальною функціональністю (див. табл. 1.1).

Таблиця 1.1 – Інтереси відносно системи

№	Зацікавлена особа	Обмеження	Інтереси
1	Школи	Будуть взаємодіяти лише з надійною системою.	Масове проведення онлайн-занять.
2	Розробник	Терміни реалізації та бюджет.	Розробка масштабується. Система відповідає критеріям якості.
3	Учні	Будуть взаємодіяти лише зі зрозумілим та адаптивним інтерфейсом.	Відвідування онлайн-занять, виконання домашнього завдання.
4	Інвестор	Не впевнений у надійності та вмілості команди.	Спланувати процес розробки та передбачити розвиток проекту.

Час і бюджет – головні обмеження розробника, таким чином потрібно відокремити пріоритетні задачі та цілі для того, щоб акцентувати свою увагу на

них у процесі розробки та сформувані стійкий процес розробки для уникнення критичних ситуацій.

1.3 Середовище реалізації

Для створення серверної частини додатку (back-end) буде використана платформа Node.js [5] та її фреймворк Express [5] та мова JavaScript для її реалізації; для створення клієнтської веб-частини (front-end) буде використана JavaScript-бібліотека для створення користувацьких інтерфейсів ReactJS [6]; для зберігання даних буде використана документо-орієнтована СКБД MongoDB [4]; для розробки мобільного додатку для пристроїв Apple під керуванням операційної системи IOS буде використане середовище розробки Expo та мова програмування React Native [7].

2 ПОСТАНОВКА ЗАДАЧИ

2.1 Постановка мети

Основною метою є реалізація робочої системи, яка дозволить викладачам створювати уроки, об'єднувати їх у курси, надавати доступ окремим учням або цілим групам учнів, а також проводити зручне оцінювання виконаних робіт; з іншого боку, учням буде дозволено переглядати доступні курси, завантажувати виконані роботи у систему, слідкувати за успішністю та контролювати терміни здачі завдань.

Сервіс повинен мати багаторівневу архітектуру, бути багат шаровим, масштабуватися, на клієнтському рівні мати веб-клієнт та мобільний додаток, мати можливість резервного копіювання даних та надавати захист персональної інформації користувачів та створеного ними контенту.

2.2 Основний функціонал системи

Система повинна включати у себе наступні головні функції:

а) взаємодія з користувачем (викладач):

- 1) реєстрація нового користувача
- 2) авторизація користувача
- 3) взаємодія з групами учнів (створення групи, додавання та видалення учня з групи, додавання курсу у групу)
- 4) взаємодія з уроками (створення, редагування, видалення, перегляд, додавання до курсу)
- 5) взаємодія з курсами (створення, редагування, видалення, перегляд, додавання у групу, публікація)

- б) взаємодія з виконаними роботами (перегляд, оцінювання)
- б) взаємодія з користувачем (учень):
 - 1) реєстрація нового користувача
 - 2) авторизація користувача
 - 3) взаємодія з курсами (перегляд, сортування)
 - 4) взаємодія з уроками (перегляд, завантаження виконаного завдання)
 - 5) перегляд успішності
 - 6) перегляд термінів здачі завдань
- в) взаємодія з іншими системами
 - 1) робота системи з мобільним пристроєм на ОС IOS

2.3 Допущення та залежності

Для коректної роботи усіх частин проекту наведений список допущень:

- а) користувач має постійний доступ до мережі Інтернет;
- б) користувач мобільного додатку має модель iPhone не нижче 7 з операційною системою IOS не нижче 12 покоління;
- в) користувач використовує браузер з підтримкою та дозволом на виконання JavaScript [8] модулів.

Також існують наступні залежності:

- а) користувачі мають базові вміння користування комп'ютером та веб-браузером;
- б) користувачі готові надати доступ до обробки та використання персональної інформації;

2.4 Релізи

Даний проект було поділено на декілька релізів. Необхідний функціонал для випуску першого релізу:

а) Back-end частина має:

- 1) реєстрація та авторизація користувачів типу «викладач» та «учень»
- 2) захист даних
- 3) API для взаємодії з курсами, уроками та групами зі сторони викладача та учня

б) Front-end частина має:

- 1) форми реєстрації та авторизації користувачів типу «викладач» та «учень»
- 2) сторінку з доступними курсами (учень) та створеними курсами (викладач)
- 3) сторінку з особистою статистикою (учень)
- 4) сторінку з термінами здачі завдань (учень)
- 5) сторінку створення/редагування курсу (викладач)
- 6) сторінку створення/редагування уроку (викладач)

в) Мобільний додаток має:

- 1) форму реєстрації/авторизації (учень)
- 2) сторінку з термінами здачі завдань (учень)
- 3) функцію повідомлень про найближчі терміни (учень)

Наступні релізи будуть мати:

- а) виправлення технічних недоліків програми
- б) покращення візуальної складової
- в) додання сервісу для комунікації викладача та учнів
- г) додання сервісу відправки/отримання повідомлень різних типів

д) додання інструментів для створення інтерактивних уроків різних типів

2.5 Користувацькі обмеження

Для коректного використання даного продукту та запобігання непорозуміння з боку користувачів, введено деякі користувацькі обмеження:

- а) користувач повинен пам'ятати дані для авторизації (пошта та пароль);
- б) користувач повинен авторизуватися для користування системою;
- в) користувач повинен дати згоду на використання деякої персональної інформації платформою

3 АРХІТЕКТУРА ПРОЕКТУ

3.1 Загальна архітектура проекту

Для написання проекту була обрана архітектура «клієнт-сервер», яка має ряд переваг, а саме: легка масштабованість, гнучкість та надійність, що є необхідним для написання проекту такого типу, оскільки вона надає можливість реалізувати один або декілька незалежних серверів, які безпосередньо будуть взаємодіяти з базою даних, та мати необмежену кількість серверів, які будуть спілкуватися з серверами.

Діаграма розгортання зображена на рисунку 3.1.

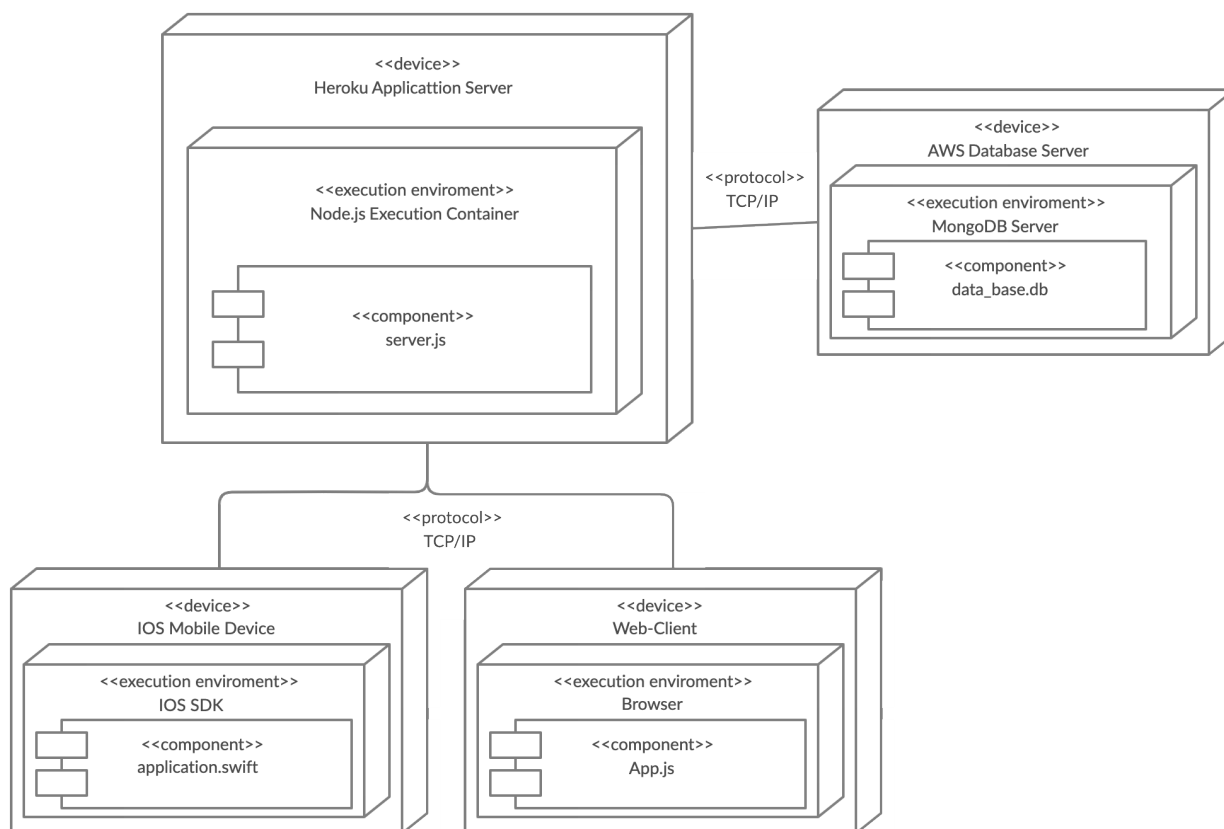


Рисунок 3.1 – Діаграма розгортання

На рисунку можна побачити сервер, який взаємодіє з базою даних, та два клієнти: веб-клієнт та мобільний-клієнт. Слід зазначити, що кожен з компонентів спілкується з сервером за допомогою TCP/IP протоколу.

На Use-Case діаграмі (див. рис. 3.2) системи зображено можливості різних користувачів, поділених на ролі (викладач та учень), взаємодіяти з системою.



Рисунок 3.2 – Use-case діаграма

У програмі передбачено дві ролі користувачів: викладач та учень.

Обидва типи користувачів можуть увійти у систему (вхід у систему передбачає отримання токена, що генерує спеціальний сервіс – Json Web Token

[3], який надалі буде валідуватися при подальших викликах API з серверної частини).

Викладач має змогу створювати групи учнів, після чого персонально додавати кожного учня до вибраної групи; створювати курси та уроки, після чого по бажанню публікувати їх у відкритий доступ; додавати урок до вибраного курсу; а також переглядати виконані завдання та за наявності можливості їх перевіряти.

Учень у свою чергу має можливість переглядати уроки та курси, а також завантажувати виконані завдання, переглядати найближчі терміни здачі завдань, а також переглядати статистику власної успішності.

3.2 Архітектура та проектування Back-end серверу

Для написання Back-end серверу була використана багаторівнева архітектура проекту (див. рис. 3.3).

Для реалізації цієї частини проекту було обрано використовувати платформу Node.JS та фреймворк Express для створення TCP серверу. Дана технологія була обрана через оптимально реалізований неблокуючий ввід/вивід, що дає змогу найшвидше вирішувати I/O-bounded задачі. Для розгортання була обрана платформа Heroku, яка дає можливість дуже швидко та просто розгорнути серверний додаток на великій кількості технологій, зокрема на Node.JS [5].

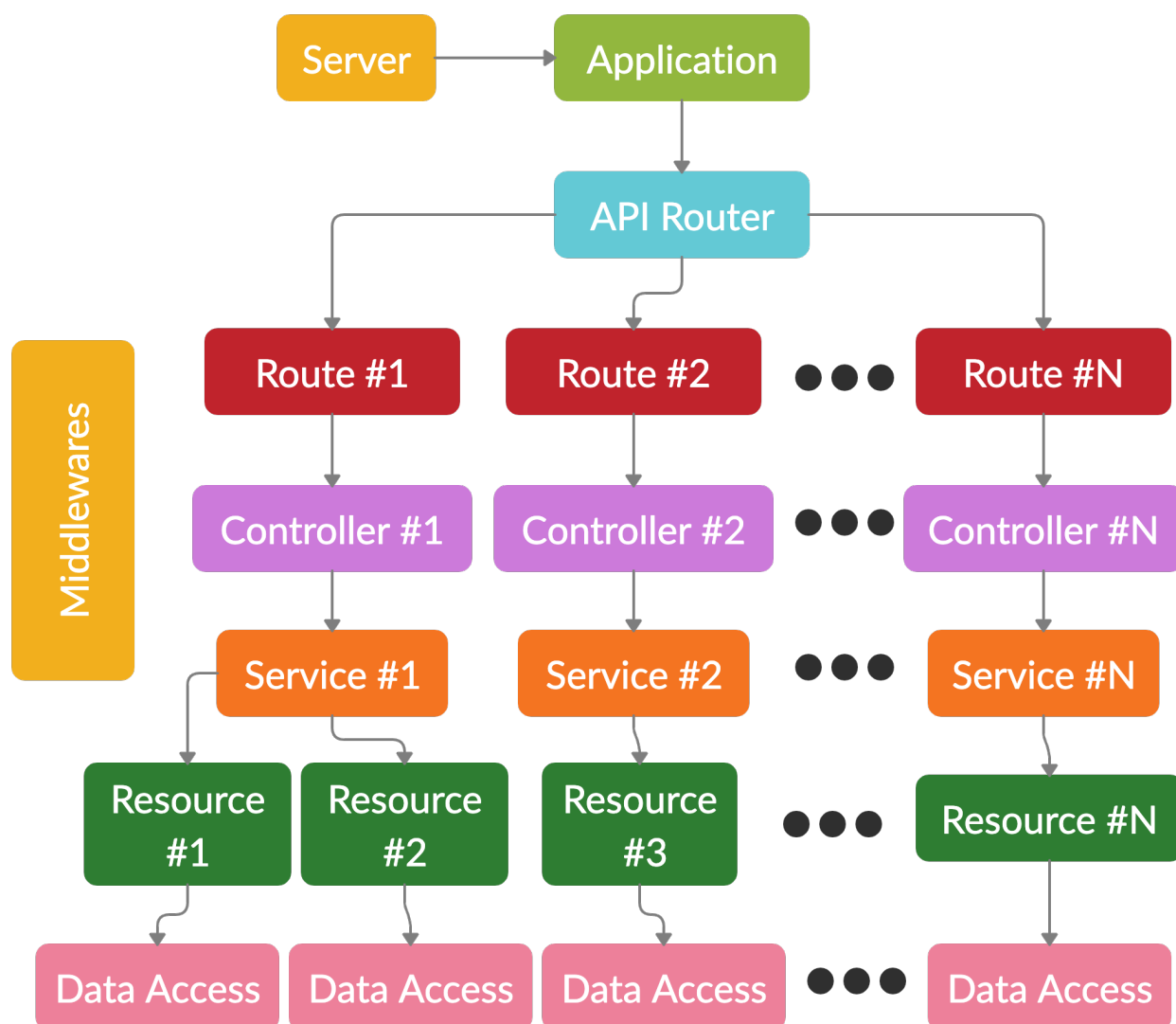


Рисунок 3.3 – Архітектура серверної частини проекту

Також у цій частині використовується багаторівнева архітектура, що складається з наступних рівнів:

а) API Router та Route – рівень взаємодії з веб-сервером, надає можливість клієнтам системи спілкуватися з сервером за допомогою кінцевих точок (endpoint) та передає запит до наступного рівня - контролера, після чого повертає клієнтові певний результат отриманий від контролера.

б) Controller – рівень обробки запитів, перевіряє запити на коректність, «розпаковує» запит, отриманий з попереднього рівня, та перетворює його у відповідні вхідні параметри для наступного рівня – сервісу, у разі виникнення помилок, формує відповідний звіт для відправки клієнту через роутер, в іншому

випадку перетворює результат у належний формат для повернення клієнтові через роутер.

в) Service – рівень бізнес логіки, виконує обробку даних, отриманих контролером, має можливість за допомогою наступного рівня – ресурсу, звернутися до бази даних задля отримання певної інформації, але не зобов'язаний завжди робити це.

г) Resource – рівень взаємодії з базою даних, надає певний функціонал доступу к даним, що зберігаються у базі даних.

Окремим компонентом системи є Middleware (наскрізний функціонал), який надає функції для автентифікації запитів та глобальної обробки помилок та виключних ситуацій, які виникають під час роботи системи.

3.3 Архітектура та проектування бази даних

У ролі бази даних було обрано документо-орієнтовану NoSQL СКБД під назвою MongoDB [4], яка зберігає дані у вигляді документів, дуже схожих на за форматом на JSON. Основними перевагами цієї бази даних є легка масштабованість, висока продуктивність, проста установка та наявність гнучких схем для зберігання неструктурованих та комплексних даних.

Розгортання бази даних у хмарному сховищі AWS відбувається за допомогою проміжного сервісу Mongo Atlas [4], який дозволяє візуалізувати дані, проводити резервне копіювання даних, контролювати доступ до бази, у ручному режимі редагувати вміст документів та багато додаткового функціоналу щодо керування базою.

База даних проектується методом опису моделі документа безпосередньо у коді програми, модель представляє собою JSON документ з атрибутами, для кожного атрибута можна визначити його тип, зазначити обов'язковість та навести інші надбудови.

Взаємозв'язок з базою відбувається за допомогою спеціального БД-клієнту для платформи Node.JS – Mongoose [4].

Приклад збереження даних у БД можна побачити на рисунку 3.4.

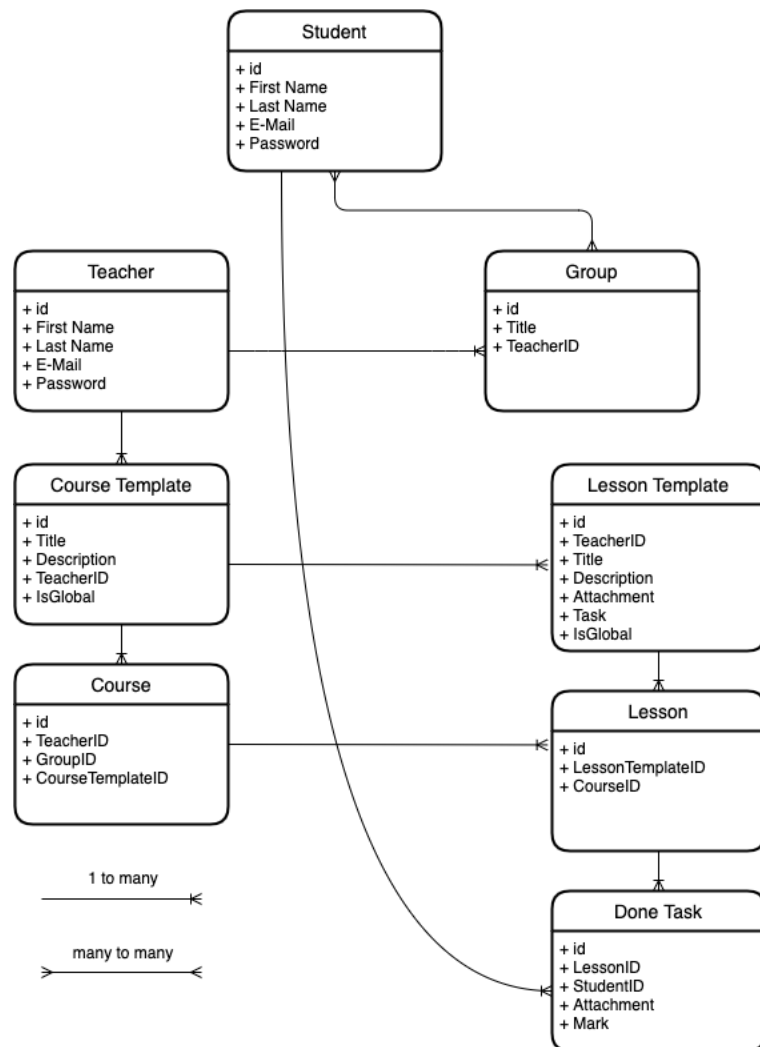


Рисунок 3.4 – Діаграма моделі даних

База має сім основних документів: «Студенти», «Викладачі», «Групи», «Шаблони курсів», «Курси», «Шаблони уроків», «Уроки», «Виконані завдання».

Документ «Студенти» зберігає інформацію про зареєстрованих студентів, а саме: ідентифікатор, ім'я, прізвище, адресу електронної пошти та пароль.

Документ «Викладачі» зберігає повністю аналогічну інформацію щодо викладачів. Користувачі були розділені на два документи задля оптимальності

деяких операцій пошуку у базі, а також через різні права доступу до деяких сутностей та різний функціонал зі сторони клієнту.

Документ «Групи» зберігає інформацію про групи студентів під керівництвом викладача, тож кожна група має свій ідентифікатор, ідентифікатор викладача-власника, та масив ідентифікаторів студентів-учасників.

Для створення будь-якого курсу викладач повинен спочатку створити його шаблон. Тож маємо два документи у базі даних для цього. Тож документ «Шаблони курсів» має ідентифікатор, назву, опис, ідентифікатор викладача-власника, а також бінарне поле, яке відповідає за те, знаходиться шаблон у глобальному доступі чи ні. Документ «Курси» зберігає реальні курси, створені за певним шаблоном, та призначені для певної групи учнів, тож він складається з ідентифікатору, ідентифікатору викладача-власника, ідентифікатора групи та ідентифікатор шаблону.

Уроки створюються аналогічним способом, за допомогою шаблонів. Документ «Шаблон уроку» складається з ідентифікатору, ідентифікатору викладача-власника, назви, опису, прикріплених файлів, доманього завдання, та аналогічного бінарного поля, що відповідає за доступ. Документ «Уроки» має ідентифікатор, ідентифікатор викладача-власника та ідентифікатор відповідного шаблону уроку, а також ідентифікатор курсу.

Останньою сутністю є документ «Виконані завдання» який має: ідентифікатор, ідентифікатор уроку, ідентифікатор студенту, прикріпленні файли та оцінку.

4 ОПИС ПРОГРАМНИХ РІШЕНЬ

4.1 Back-end сервер

Back-end сервер був реалізований на базі найпопулярнішої JavaScript [8] платформи Node.JS [4].

Можливість доступу клієнтів до сервера було реалізовано за допомогою REST API на базі фреймворку Express. Запити користувачів у спеціалізованому форматі відправляються на сервер по спеціально призначених маршрутах. Після отримання запиту, рівень контролеру перевіряє дані на коректність та відправляє на обробку до рівня бізнес логіки. Рівень бізнес-логіки автоматизує усі необхідні процеси та у разі потреби звертається до даних у сховище (в нашому випадку – бази даних).

Для реалізації такої структури компонентів було реалізовано принцип під назвою Dependency Injection [1]. Через використання цього принципу було знижено рівень зв'язності компонентів архітектури, що зробило роботу з окремим компонентом більш гнучку та комфортну.

Механізм авторизації був реалізований за допомогою спеціальної бібліотеки JWT (JSON Web Token) [5], що дозволяє генерувати крипостійкі та захищені токени, які зберігають інформацію щодо ідентифікації поточного користувача.

Для взаємодії з базою даних був використаний спеціальний клієнт Mongoose [4], через який можна проводити операції з поточною БД, що знаходиться у хмарному сховищі.

4.2 Front-end частина

У даній курсовій роботі користувацький веб-інтерфейс було розроблено за допомогою відповідної бібліотеки проектування гнучких інтерфейсів ReactJS [6], що дозволяє оновлювати контент на сторінці без додаткових дій оновлення веб-сторінки у браузері.

Веб-клієнт повністю відповідає за відображення користувацького інтерфейсу та усієї інформації, що поступає з боку серверу. Для отримання даних веб-клієнт відправляє спеціальні запити із певними параметрами, які описує API серверної частини.

Веб-клієнт можна використовувати у таких браузерах, як: Safari, Mozilla Firefox, Opera, Edge та усі браузери, які реалізовано на платформі Chromium. Також веб-клієнт обов'язково повинен мати доступ до мережі Інтернет.

Веб-клієнт реалізовано таким чином, що контент, який він відображає користувачу залежить від поточного URL адреси.

У системі передбачено два типи акаунтів: викладачі та учні. Під час реєстрації та авторизації у користувача є можливість обрати необхідну роль (див. рис. 4.1). Тож на одну пошту можна зареєструвати два облікових записи для різних ролей.

The image shows a login interface on a dark blue background. A white card contains the 'Sign In as' section with two buttons: 'Student' (blue) and 'Teacher' (white). Below these are two input fields: an email field with the placeholder 'student@gmail.com' and a password field with masked characters. A blue 'SIGN IN' button is positioned below the inputs. At the bottom of the card, there are two links: 'Forgot password?' and 'Create new account'.

Рисунок 4.1 – Форма авторизації

Після вдалої авторизації користувач має можливість навігації по платформі за допомогою меню (див. рис. 4.2). Користувач-учень має такі розділи у меню: «Профіль», «Курси», «Терміни» та «Статистика». Користувач-викладач у свою чергу має відповідні розділи: «Профіль», «Курси», «Групи», «Оцінювання».

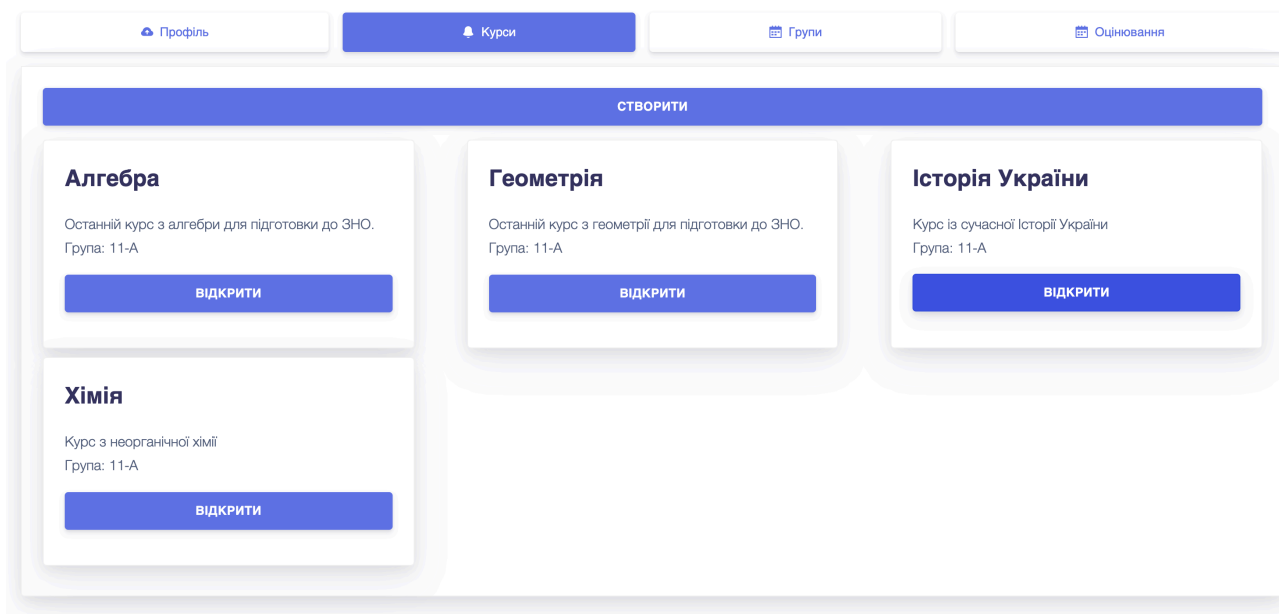


Рисунок 4.2 – Меню навігації та розділ «Курси»

Кожна сторінка надає відповідний функціонал користувачу для взаємодії з платформою.

Взаємодія веб-додатку с сервером відбувається за допомогою спілкування з REST API серверу шляхом відправки запитів та отримання результату на них.

Користувацький інтерфейс підтримує англійську та українську локалізацію. Для її реалізації був використаний спеціальний модуль React-Localization [5].

4.3 Мобільний пристрій

У даному курсовому проекті мобільний додаток написаний за допомогою платформи React Native [7], яка трансформує універсальний код окремо під кожен платформу (IOS та Android), із урахуванням усіх особливостей кожної з операційних систем.

Функціонал додатку можна поділити на дві складові:

а) першою складовою є авторизація та реєстрація

б) другою є можливість перевіряти найближчі терміни здачі завдань та переходити за посилання до умови завдання (див. рис. 4.3)

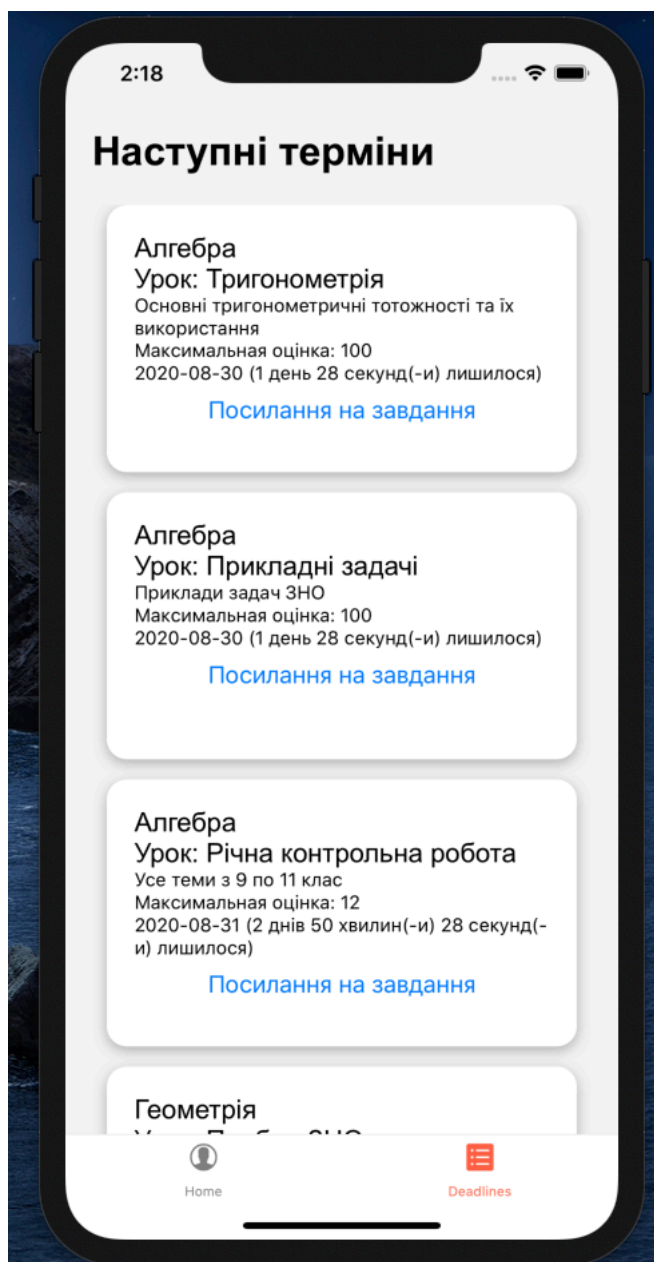


Рисунок 4.3 – Основний розділ мобільного додатку

Також, для використання мобільного додатку, треба бути зареєстрованим користувачем. Також користування мобільним додатком можливо лише користувачам-учням, оскільки функціонал, що надає мобільний додаток на даному етапі проектування актуальний лише для користувачів-учнів.

ВИСНОВКИ

У результаті виконання даної курсової роботи була розроблена та реалізована робоча системи, яка дозволяє викладачам створювати уроки, об'єднувати їх у курси, надавати доступ окремим учням або цілим групам учнів, а також проводити зручне оцінювання виконаних робіт; з іншого боку, учні мають змогу переглядати доступні курси, завантажувати виконані роботи у систему, слідкувати за успішністю та контролювати терміни здачі завдань.

Back-end частина проекту була реалізована на основі платформи Node.JS та її фреймворку Express. Веб-клієнт було реалізовано за допомогою бібліотеки створення інтерфейсів ReactJS [6]. Мобільний додаток для платформи IOS був реалізований за допомогою мови програмування React Native [7] та набору інструментів IOS SDK [7].

Сервіс має багаторівневу архітектуру, є багат шаровим та легко масштабується, клієнтський рівень складається з веб-клієнту та мобільного додатку, має можливість резервного копіювання даних та надає захист персональної інформації користувачів та створеного ними контенту.

У ході виконання роботи та реалізації системи також були накопичені навички розробки архітектури систем, розробки серверної архітектори, моделювання NoSQL бази даних та побудови зручного користувацького інтерфейсу.

Під час розробки даної системи були успішно реалізовані усі вимоги, сформульовані у постановці задачі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Макконнелл С. Досконалий код [Текст] . Макконнелл С. -СПб .: Питер, 2007. -896 с.
2. Google Classroom [Електронний ресурс] – Режим доступу до ресурсу: <https://classroom.google.com/>
3. Json Web Token Library [Електронний ресурс] – Режим доступу до ресурсу: <https://jwt.io/>
4. MongoDB Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.mongodb.com/guides/>
5. Node.js v13 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/docs/latest-v13.x/api/>
6. React Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://reactjs.org/docs/getting-started.html>
7. React Native Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://reactnative.dev/>
8. The Modern JavaScript Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://javascript.info/>

ДОДАТОК А

Vision and Scope Document

для

Програмна система для віддаленого навчання дітей в середній школі (5-11 клас)

Версія 1.0 затверджена

Підготував ст. гр. ПЗП-17-2

Руденко Дмитро Олександрович

Харківський національний університет радіоелектроніки

07.04.2020

1 Бізнес вимоги

1.1 Передумови

У сучасному світі під впливом пандемії COVID-19 тенденції активно рухаються у напрямку дистанційних технологій та сервісів. Галузь освіти одна з найбільших, що потрапили під цей вплив.

Онлайн-платформи для навчання активно використовуються студентами в усьому світі, але серед школярів навчальної та середньої школи цей підхід поки що не ціниться у повній силі. Цей факт обумовлений тим, що більшість у цьому віку лише починають знайомитися з комп'ютером та взаємодією з ним.

1.2 Можливості для бізнесу

Завдяки сучасним технологіям існує можливість створити платформу, якою можна користуватися з мінімальним досвідом користування комп'ютерними пристроями, завдяки інтуїтивно зрозумілому інтерфейсу та яка буде мати максимальний рівень залучення уваги учнів завдяки гейміфікації та інтерактивності завдань.

Критерії успіху даної системи базуються на тому, що система надає можливість у разі необхідності повністю забезпечити дистанційне навчання учнів шкіл, а також забезпечує можливість проведення дистанційних занять з репетитором або персональним викладачем.

Проаналізувавши ці дані, можна зазначити, що є велика необхідність у розробці продукту, який забезпечив би школи у разі необхідності зручним

ресурсом дистанційної освіти, а також дати можливість репетиторам проводити заняття онлайн та розширювати аудиторію.

1.3 Бізнес-цілі та критерії успіху

BO-1: задовільнити попит на платформи дистанційної освіти серед шкіл

BO-2: надати можливість репетиторам проводити заняття онлайн

BO-3: надати можливість викладачам створювати та поширювати навчальний контент

SC-1: отримати щонайменше 5000 активних користувачів-викладачів

SC-2: задіяти щонайменше 200 шкіл у використанні платформи

SC-3: наповнити глобальну бібліотеку щонайменше 100 курсами з різних предметів

1.4 Користувачі та потреби ринку

Дуже велику цінність представляють користувачі-викладачі, але основними користувачами сервісу будуть користувачі-учні. Викладачам буде запропоновано створення та розповсюдження уроків та курсів у цифровому форматі. Після чого учні будуть кожного дня «споживати» навчальні матеріали, створені викладачами. В обох випадках основною потребою користувачів є багатофункціональність та зручність використання інтерфейсом, оскільки більшість викладачів та учнів мають не великий досвід керуванням комп'ютера.

1.5 Бізнес ризики

Основна привабливість платформи базується на можливості створювати контент викладачами, тож основним бізнес-ризиком є втрата частини контенту, навіть дуже малої частини, оскільки це дуже сильно підірве репутацію платформи, що буде супроводжуватись значним зменшенням кількості активних користувачів.

Також, великим ризиком можуть стати конкуренти, оскільки сфера шкільної онлайн освіти ще зовсім не розвинена і має великий потенціал для розвитку, світові ІТ-гіганти можуть забезпечити набагато більш функціональний сервіс за рахунок наявності великої кількості ресурсів. Прикладом може послужити корпорація Google, яка має схожий сервіс Classroom, який може бути не просто агрегацією основних сервісів Google, а набагато функціональним продуктом.

2 Бачення рішення

2.1 Уявлення вкладу

Головною метою є реалізація робочої системи, яка дозволить викладачам створювати уроки, об'єднувати їх у курси, надавати доступ окремим учням або цілим групам учнів, а також проводити зручне оцінювання виконаних робіт; з іншого боку, учням буде дозволено переглядати доступні курси, завантажувати виконані роботи у систему, слідкувати за успішністю та контролювати терміни здачі завдань.

2.2 Основні характеристики

Система буде мати наступні можливості:

FE-1: Реєстрація користувачів двох типів (учень та викладач) з різними можливостями взаємодії з платформою;

FE-2: Створення курсів, уроків та навчальних матеріалів та можливість публікації їх у відкритий доступ іншим вчителям;

FE-3: Виконання та завантаження виконаних робіт учнями у систему;

FE-4: Оцінювання виконаних робіт викладачем;

FE-5: Перегляд успішності з різних курсів учнем;

FE-6: Перегляд термінів здачі завдань учнем;

2.3 Припущення та залежності

AS-1: Учень може належати до декількох груп під керівництвом різних викладачів.

AS-2: Усі учні у групі та викладач розуміють та можуть створювати контент хоча б однією спільною мовою.

DE-1: Для функціонування веб-додатку та мобільного додатку, користувач обов'язково повинен мати доступ до мережі Інтернет. Ця необхідність виникає через користування засобами API, які потребують підключення до мережі.

3 Область застосування та обмеження

3.1 Сфера первинного випуску

Проект складається з декількох частин (Front-end, Back-end та Mobile), кожна з яких у свою чергу складається з різних компонентів. Нижче наведено опис кожної з частин:

а) Back-end частина є основною частиною проекту, оскільки саме вона повністю забезпечує обробку даних та взаємодію зі сховищем, де ці дані зберігаються (в даній реалізації проекту – у базі даних). Також ця частина спілкується з іншими частинами, що відповідають за представлення результату виконання бізнес-логіки кінцевому користувачу.

б) Front-end частина надає користувачу елементи інтерфейсу для взаємодії з Back-end частиною, а також відображає усю необхідну користувачу інформацію отриману у результаті спілкування з Back-end частиною.

в) Mobile частина, як і Front-end частина, відповідає за представлення інформації користувачу, але на відміну від Front-end частини, вона робить це не у повному обсязі та надає користувачу спрощений функціонал.

3.2 Сфера подальшого випуску

Наступні випуски будуть мати:

- 1) Більшу кількість інструментів для створення уроків.
- 2) Можливість спілкування у системі між учнями та викладачем.
- 3) Більш гнучка статистика успішності.

3.3 Обмеження та виключення

- 1) Контент, створений користувачами-викладачами не повинен порушувати чинне законодавство та закони у країні, де цей контент було створено або розповсюджено.
- 2) Доступ до контенту, створеному користувачем-викладачем повинен надаватися лише користувачам, що були обрані тим, хто цей контент створив.

4 Бізнес контекст

4.1 Профілі зацікавлених сторін

Обмеження	Основні інтереси	Позиції	Основна цінність	Зацікавлена сторона
Будуть взаємодіяти лише з надійною системою.	Масове проведення онлайн-занять.	Зацікавлена сторона	Успішне проведення заняття та оцінювання учнів у дистанційному форматі	Школи
Терміни реалізації та бюджет.	Розробка масштабується. Система відповідає критеріям якості.	Творець	Отримання позитивного відгуку від користувачів проекту	Розробник
Будуть взаємодіяти лише зі зрозумілим та	Відвідування онлайн-занять, виконання домашнього завдання.	Зацікавлена сторона	Отримання нових знань та проходження оцінювання у	Учні

адаптивним інтерфейсом.			дистанційному форматі	
Не впевнений у надійності та вмілості команди.	Спланувати процес розробки та передбачити розвиток проекту.	Зацікавлена сторона	Отримання доходу від інвестування у проект	Інвестор

4.2 Пріоритети проекту

Ступінь свободи	Ключовий фактор	Обмеження	Параметри
У першому релізі повинні працювати 65-75% усього функціоналу.			Функціонал
	Перший реліз повинен з'явитися до 15.06.2020		Графік
		Бажано створення системи з нульовими матеріальними витратами	Витрати
		Максимальна кількість розробників продукту – лише один студент.	Персонал

4.3 Середовище експлуатації

Розроблена платформа буде доступна для використання усім користувачам таких веб-браузерів, як: Google Chrome, Mozilla Firefox, Opera, Edge, Safari; та на пристроях iPhone вище 7 покоління.

Інструментами для розробки будуть виступати:

Back-end: Node.JS, Express;

Front-end: HTML5, CSS, ReactJS

Mobile: React Native, IOS SDK