

МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

На тему: Розробка інтерактивного дизайну освітнього додатку на платформі
Андроїд

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи ЦДБ-20-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Технології цифрового дизайну»



(підпис)

Плясун Іван Іванови
(прізвище, ім'я, по батькові)

Науковий керівник



(підпис)

Омельчук Антон
Анатолійович, к.т.н.
(прізвище, ім'я, по батькові,
науковий ступінь, вчене звання)

Керівник освітньо-
професійної програми



(підпис)

Ратушняк Тетяна
Володимирівна, к.ф.-м.н.,
доцент
(прізвище, ім'я, по батькові, науковий
ступінь, вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних
технологій і систем протокол від «22» травня 2024 № 15

Завідувач кафедри



(підпис)

Філоненко Михайло
Миколайович, к.ф.-м.н.,
доцент
(прізвище, ім'я, по батькові,
науковий ступінь, вчене звання)

Реєстрація № 51

31.05.2024

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту
(до захисту, на доопрацювання)

22.05.2024

(дата)



(підпис)

Результат захисту

4 (добре)
(оцінка)

12.06.2024

(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

Члени комісії:



(підпис)

М.М. Філоненко



(підпис)

С.М. Грищенко



(підпис)

В.О. Ніжегородцев

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	22		12.06.2024 Франчук В.М.  Філоненко М.М.  Грищенко С.М.  Ніжегородцев В.О.  Суховерхівка Л.П. 
2. Дотримання вимог щодо оформлення	від 0 до 10	8		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	3		
4. Рецензент	від 0 до 5	3		
РАЗОМ БАЛІВ	від 0 до 50	36		

Оцінювання кваліфікаційної роботи науковим керівником - 34 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підписи)
70	C	4 (добре)	12.06.2024	Франчук В.М.  Філоненко М.М.  Грищенко С.М.  Ніжегородцев В.О.  Суховерхівка Л.П. 

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Технології цифрового дизайну»

Затверджую

Завідувач кафедри Філоненко М.М.

«23» жовтня 2023

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Плясуну Івану Івановичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Розробка інтерактивного дизайну освітнього додатку на платформі Андрюїд.
Затверджена наказом ректора від «20» жовтня 2023 № 1411
2. Строк здачі закінченої роботи «21» травня 2024
3. Мета роботи дослідити існуючі аналоги та розробити власне рішення.
Об'єкт дослідження розробка програмного забезпечення аналог освітнього додатку на платформі Андрюїд.
Предмет дослідження процес освіти на дистанційному рівні.
Орієнтовна база дослідження додатки на платформі Андрюїд.
4. Вихідні дані до роботи наукові публікації на відповідну тематику.
5. Перелік графічного матеріалу UML діаграми до застосунку, схеми принципів роботи застосунку
6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)
 1. ХАРАКТЕРИСТИКА РОЗРОБКИ ОСВІТНІХ ДОДАТКІВ
 - 1.1. Теоретичні основи інтерактивного дизайну
 - 1.2. Визначення інтерактивного дизайну
 - 1.3. Основні принципи та методики інтерактивного дизайну
 - 1.4. Особливості інтерактивного дизайну в освітніх додатках
 - 1.5. Аналіз існуючих рішень
 - РОЗДІЛ 2. ПРОЄКТУВАННЯ ОСВІТНЬОГО ДОДАТКУ ДЛЯ ПЛАТФОРМИ АНДРЮїД
 - 2.1. Аналіз сильних та слабких сторін існуючих додатків
 - 2.2. Розробка каркасів (Wireframes) та макетів (Mockups)
 - 2.3. Проектування інтерактивного дизайну
 - РОЗДІЛ 3. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ЗАСТОСУНКУ
 - 3.1. Створення користувацьких сценаріїв (User Stories)
 - 3.2. Організація та проведення користувацького тестування
 - 3.3. Аналіз результатів тестування

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 15.10.2023	До 15.10.2023
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	29.12.2023	29.12.2023
3	II розділ (аналіз експериментальної частини про методики досліджень)	16.02.2024	16.02.2024
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	16.04.2024	16.04.2024
5	Висновки	30.04.2024	30.04.2024
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	10.05.2024	10.05.2024
7	Отримання відгуку керівника	30.05.2024	30.05.2024
8	Отримання зовнішньої рецензії	03.06.2024	03.06.2024

8. Дата видачі завдання «23» жовтня 2023

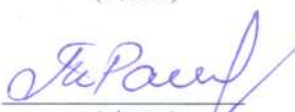
9. Керівник кваліфікаційної роботи

10. Керівник освітньо-професійної програми

11. Завдання до виконання отримав
здобувач вищої освіти


(підпис)

Омельчук А.
(прізвище, ініціал)


(підпис)

Ратушняк Т.
(прізвище, ініціал)


(підпис)

Плясун І.І.
(прізвище, ініціал)

АНОТАЦІЯ

"Розробка інтерактивного дизайну освітнього додатку на платформі Андроїд"

Мета роботи полягає в дослідженні теми та розробці аналогу освітнього додатку на платформі Андроїд

Завданнями роботи є розробити аналог освітнього додатку.

Предметом дослідження є процес освіти на дистанційному рівні

Об'єктом дослідження є розробка програмного забезпечення аналог освітнього додатку на платформі Андроїд

Одержані результати можуть бути використанні на теперешній час. Оскільки на даний час існує велика потреба переведення на дистанційну форму навчання.

Кваліфікаційна робота містить 63 сторінки 1 таблицю, 11 рисунків, 30 посилань та 1 додаток.

Ключові слова: освіта, система, онлайн навчання, розробка.

ABSTRACT

"Development of interactive design of an educational application on the Android platform"

The purpose of the work is to study the topic and develop an analog of an educational application on the Android platform

The tasks of the work are to develop an analog of an educational application.

The subject of research is the process of education at the distance level

The object of research is the development of software analogous to an educational application on the Android platform

The results obtained can be used at the present time. Since there is currently a great need for distance learning.

The qualification work contains 63 pages, 1 table, 11 figures, 30 references and 1 appendix.

Keywords: education, system, online learning, development.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ХАРАКТЕРИСТИКА РОЗРОБКИ ОСВІТНІХ ДОДАТКІВ	4
1.1. Теоретичні основи інтерактивного дизайну	4
1.2. Визначення інтерактивного дизайну	7
1.3. Основні принципи та методика інтерактивного дизайну.....	8
1.4. Особливості інтерактивного дизайну в освітніх додатках	10
1.5. Аналіз існуючих рішень	15
РОЗДІЛ 2. ПРОЄКТУВАННЯ ОСВІТНЬОГО ДОДАТКУ ДЛЯ ПЛАТФОРМИ АНДРОЇД.....	26
2.1. Аналіз сильних та слабких сторін існуючих додатків	26
2.2. Розробка каркасів (Wireframes) та макетів (Mockups)	38
2.3. Проектування інтерактивного дизайну	41
РОЗДІЛ 3. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ЗАСТОСУНКУ	45
3.1. Створення користувацьких сценаріїв (User Stories).....	45
3.2. Організація та проведення користувацького тестування	47
3.3. Аналіз результатів тестування.....	51
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТКИ.....	58

ВСТУП

Мобільні додатки дозволяють продовжувати навчання в районах бойових дій і навіть зон катастрофи, демонструючи яскраве втілення спадкоємності.

Безумовно, діти люблять користуватися навчальними програмами. Вони заряджають енергією та розважають. Вчителям може бути важко переконати учнів бути уважними на уроці, але вони рідко намагаються змусити їх використовувати комп'ютер. Заняття фізкультурою проходять легко, тому що можна вийти на місце учня і зрозуміти його. На таких заняттях все, що вам потрібно зробити, це зробити заняття більш цікавими, що спонукатиме студентів автоматично зацікавитися заняттям. Як правило, ці заняття також є розважальними. Але є студенти, які не завжди можуть бути залучені до лекції, вони можуть зацікавитися певним предметом через додаток.

Використання освітніх програм може допомогти учням більше зацікавитися предметами, яких вони могли б уникати.

Мета роботи полягає в дослідженні теми та розробці аналогу освітнього додатку на платформі Андроїд

Завданнями роботи є розробити аналог освітнього додатку.

Предметом дослідження є процес освіти на дистанційному рівні

Об'єктом дослідження є розробка програмного забезпечення аналог освітнього додатку на платформі Андроїд

Одержані результати можуть бути використанні на теперешній час. Оскільки на даний час існує велика потреба переведення на дистанційну форму навчання.

РОЗДІЛ 1. ХАРАКТЕРИСТИКА РОЗРОБКИ ОСВІТНІХ ДОДАТКІВ

1.1. Теоретичні основи інтерактивного дизайну

Юзабіліті навчального середовища в рамках цифрових технологій потребує належної взаємодії між користувачем та інтерфейсом мобільного додатку. Згідно з Муслімом та ін. (2019), користувацький інтерфейс (UI) - це системний дисплей, який взаємодіє з операційною системою за допомогою графічних зображень, піктограм, меню та вказівного пристрою, такого як миша або трекбол [31].

Завдяки якості взаємодії, інтерфейс користувача складається з чотирьох вимірів: вимір введення або виведення (зовнішній вигляд), вимір діалогу (відчуття), технічний або функціональний вимір (доступ до інструментів і сервісів) та організаційний вимір (підтримка комунікації та спільної роботи). Крім того, інтерфейс прикрашає зовнішній вигляд і полегшує користувачеві взаємодію з системою. Привабливий, простий у використанні дисплей системи підвищить задоволеність клієнтів. Таким чином, користувацький інтерфейс повинен включати в себе стільки функцій системи, з якими знайома очікувана аудиторія користувачів [2].

Як візуальна частина платформи, користувацький інтерфейс відіграє важливу роль у забезпеченні взаємодії користувача з додатком і сприйнятті інформації користувачем. Сам користувацький інтерфейс поєднує в собі концепції візуального дизайну, дизайну взаємодії та інформаційної інфраструктури.

Переконавшись у тому, що користувацький інтерфейс задовольняє користувача, досвід користувача може визначити, наскільки легким чи складним є інтерфейс при взаємодії з елементами інтерфейсу, створеними дизайнерами інтерфейсу. Інструкції з підготовки дизайну користувацького інтерфейсу в мобільному додатку можуть відрізнятися залежно від операційної системи, яка використовується [5].

Усі продукти від Google, включно з Android, відповідають вимогам Material Design Guidelines . Ці рекомендації стосуються відображення на різних пристроях, використання зображень і символів, а також взаємодії з користувачем. Рекомендації, описані в Material Design Guidelines, є всеосяжними. Тому при розробці інтерфейсу для UNI Couse будуть обрані лише деякі з відповідних компонентів дизайну для тематичних досліджень. Нижче наведено деякі компоненти, сконцентровані в цьому тематичному дослідженні: (1) макет, кольорове оформлення та вигляд елементів керування дисплеєм. По-перше, макет - це частина дизайну інтерфейсу, яка регулює візуальні елементи на сторінці в інтерфейсі. Потім, колір є одним з найважливіших компонентів, що складають систему. Таким чином, колірний дизайн також повинен бути стратегічним. І нарешті, подання керування відображенням дозволяє користувачам взаємодіяти з даними через інтерфейс користувача [23].

Налаштування макета в інтерфейсі користувача будуть відрізнятися залежно від пристрою, який використовується для взаємодії. Наприклад, налаштування макета в комп'ютерних і мобільних додатках відрізняються через різні розміри екранів. Загальними положеннями при налаштуванні макета є вирівнювання, близькість або відстань, сітка, масштаб, а також пробіли або порожні області. Однак, додатковий розмір екрану є особливим елементом, який необхідно враховувати при створенні макета.

Елементи макета в Material Design згруповані за цими категоріями: Одиниці та вимірювання, метрики та ключові слова, структура та адаптивний користувацький інтерфейс [24].

Поєднання кольорів, яке привертає увагу користувачів, має бути розміщене стратегічно, а послідовне використання кольорів також допоможе користувачам зрозуміти додаток . Використання кольору на інтерактивному символі або кнопці привертає увагу користувача та інформує його про те, що символ виконує певну функцію . Згідно з Керівництвом з матеріального дизайну (Material Design Guidelines), принцип використання кольору для Android полягає в тому, щоб використовувати жирний колір як атракціон, оточений більш

стриманим кольором як основним фоном, а також акцентувати тіньові ефекти і світлі кольори для привернення уваги.

Дизайн-мислення - це метод, натхненний тим, як погляд дизайнера максимізує потреби споживачів [6] для створення інноваційного продукту. За словами Прадани та Ідріса (2021), дизайн-мислення - це підхід до дизайну, орієнтований на людину, який фокусується на вирішенні проблем та впровадженні інновацій. Дизайн-мислення має п'ять етапів: емпатія, визначення, ідея, прототип і тестування.

Перший етап - емпатія. На цьому етапі ми зосереджуємося на цільовому користувачеві, якому потрібна платформа. Емпатія має важливе значення для створення інноваційного продукту, оскільки знання того, що користувач думає, говорить, відчуває і робить, допоможе описати його потреби. Виділення процесу має вирішальне значення для опису потреб і проблем, з якими стикається користувач, що допоможе перейти до наступного етапу дизайн-мислення [29].

Другий етап - визначення. Етап визначення - це етап, який класифікує проблему на основі етапу співпереживання. Метод на цьому етапі - «Як би ми могли». Цей метод гарантує, що ми можемо дослідити якомога більше ідей щодо вирішення проблеми чи виклику.

Третій етап - ідеалізація. На цьому етапі формулюються ідеї, які можуть стати вирішенням проблем, отриманих на етапі визначення. Етап ідеації здійснюється шляхом оцінювання кількох креативних ідей.

Четвертий етап - створення прототипу. Прототип має на меті з'ясувати, як користувачі поведуться по відношенню до продукту. При створенні прототипів використовуються два типи: виготовлення низькоточних фреймворки та високоточні прототипи. Низькоточний фреймворк - це найпростіший дизайн, тоді як високоточний прототип - це остаточний дизайн, який використовується для тестування користувачами [15].

П'ятий етап - тестування. Цей етап тестування є життєво важливим у процесі дизайн-мислення, оскільки результати вимірюють ефективність та результативність. Результати тестування у вигляді користувацького досвіду та

зворотного зв'язку від користувачів або користувачів допоможуть оцінити продукт або додаток, який буде впроваджено. Одним з етапів тестування, що використовується в дизайн-мисленні, є юзабіліті-тестування. Юзабіліті може прояснити кожну взаємодію користувача з додатком.

Знання даної інформації забезпечує міцну основу для розробки освітнього додатку на платформі Android, охоплюючи теоретичні та практичні аспекти інтерактивного дизайну. Включення методологій, таких як Material Design Guidelines та дизайн-мислення, допомагає створити привабливий і зручний інтерфейс, що задовольняє потреби користувачів. Це сприяє кращому розумінню процесу розробки та підвищує шанси на успішне впровадження освітнього додатку.

1.2. Визначення інтерактивного дизайну

UX-дизайн — це формування досвіду використання продукту, і більша частина цього досвіду передбачає певну взаємодію між користувачем і продуктом. Але UX-дизайн — це більше, ніж дизайн взаємодії: він також передбачає дослідження користувачів (з'ясування, хто такі користувачі в першу чергу), створення особи користувача (чому та за яких умов вони будуть використовувати продукт), виконання тестування користувача та зручності використання тестування тощо.

Слова, особливо ті, що використовуються під час взаємодії, як-от написи кнопок, мають бути значущими та простими для розуміння. Вони повинні повідомляти інформацію користувачам, але не надто багато інформації, щоб перевантажити користувача.

2D: Візуальні представлення

Це стосується графічних елементів, як-от зображень, типографіки та значків, з якими взаємодіють користувачі. Зазвичай вони доповнюють слова, які використовуються для передачі інформації користувачам.

3D: фізичні об'єкти або простір

За допомогою яких фізичних об'єктів користувачі взаємодіють із продуктом? Ноутбук із мишкою чи тачпадом? Або смартфон з пальцями користувача? І в якому фізичному просторі це робить користувач? Наприклад, користувач стоїть у переповненому потязі, користуючись додатком на смартфоні, чи сидить за столом в офісі й переглядає веб-сайт? Усе це впливає на взаємодію між користувачем і продуктом [4].

4D: Час

Хоча цей вимір звучить трохи абстрактно, він здебільшого стосується засобів масової інформації, які змінюються з часом (анімація, відео, звуки). Рух і звуки відіграють вирішальну роль у візуальному та звуковому зворотному зв'язку між користувачами. Також викликає занепокоєння кількість часу, який користувач витрачає на взаємодію з продуктом: чи можуть користувачі відстежувати свій прогрес або відновити свою взаємодію через деякий час?

5D: Поведінка

Це включає механізм продукту: як користувачі виконують дії на веб-сайті? Як користувачі працюють із продуктом? Іншими словами, це те, як попередні виміри визначають взаємодію продукту. Він також включає реакції — наприклад, емоційні відповіді чи відгуки — користувачів і продукту.

Інформація даного розділу надає чітке і структуроване визначення інтерактивного дизайну в контексті UX-дизайну, підкреслюючи багатовимірний підхід до створення користувацького досвіду. Включення різних аспектів взаємодії, таких як текстові елементи, графіка, фізичний простір, час та поведінка користувача, забезпечує глибоке розуміння того, як створювати ефективні та користувацько-орієнтовані продукти. Цей підхід допомагає дизайнерам краще зрозуміти та задовольнити потреби користувачів, що є ключовим для успішного дизайну освітніх додатків на платформі Android.

1.3. Основні принципи та методики інтерактивного дизайну

1. Наочність

Видимість означає здатність користувачів бачити та розуміти доступні варіанти взаємодії з цифровим продуктом. Важливо переконатися, що всі функції та функції чітко видно користувачам. Цього можна досягти, використовуючи чіткі та описові мітки, піктограми та кнопки. Видимість допомагає користувачам зрозуміти, які варіанти доступні для них і як взаємодіяти з продуктом.

2. Консистенція

Узгодженість означає використання однакових елементів дизайну в цифровому продукті. Узгодженість допомагає користувачам зрозуміти, як взаємодіяти з продуктом, і зменшує плутанину. Дуже важливо використовувати узгоджені колірні схеми, типографіку та макет, щоб користувачі могли легко розпізнавати різні елементи продукту.

3. Картографування

Відображення стосується зв'язку між елементами керування та їхніми діями. Ви повинні переконатися, що елементи керування розміщені логічно та інтуїтивно зрозуміло. Користувачі повинні легко зрозуміти зв'язок між різними елементами керування та їхніми діями. Наприклад, якщо є кнопка для ввімкнення світла, її слід розмістити в зручному місці, яке легко знайти.

4. Зворотній зв'язок

Зворотній зв'язок – це реакція цифрового продукту на взаємодію користувача з ним. Зворотний зв'язок може бути візуальним, слуховим або тактильним. Обов'язково надішліть відгук користувачам, щоб підтвердити, що їхні дії були зареєстровані продуктом. Зворотний зв'язок допомагає зменшити невизначеність і підвищує впевненість користувачів у їх взаємодії з продуктом [24].

5. Обмеження

Обмеження стосуються обмежень, накладених на дії, які може виконувати користувач. Обмеження допомагають запобігти помилкам і скеровують користувачів до правильних дій. Обмеження можуть бути фізичними, логічними або культурними. Наприклад, логічним обмеженням може бути повідомлення

про помилку, яке з'являється, коли користувач вводить неправильну інформацію.

6. Простота

Простота означає легкість, з якою користувач може взаємодіяти з цифровим продуктом. Важливо, щоб інтерфейс був простим та інтуїтивно зрозумілим. Простий інтерфейс зменшує когнітивне навантаження на користувачів і дозволяє їм зосередитися на поставленому завданні. Важливо уникати безладу і непотрібної складності.

7. Гнучкість

Гнучкість означає здатність цифрового продукту адаптуватися до потреб різних користувачів. Дуже важливо надавати користувачам різні варіанти взаємодії з продуктом. Наприклад, ви можете надати користувачам різноманітні методи введення, як-от жести дотиком або голосові команди. Важливо надавати різні варіанти, щоб задовольнити потреби та вподобання різних користувачів [23].

Даний розділ охоплює основні принципи та методики інтерактивного дизайну, які є критичними для створення ефективного освітнього додатку на платформі Android. Принципи наочності, консистенції, картографування, зворотного зв'язку, обмежень, простоти та гнучкості забезпечують створення зручного, інтуїтивного та адаптивного інтерфейсу, який сприятиме позитивному користувацькому досвіду та підвищенню ефективності навчання. Використання цих принципів дозволяє дизайнерам створювати додатки, які відповідають потребам користувачів та забезпечують легкість і зручність взаємодії.

1.4. Особливості інтерактивного дизайну в освітніх додатках

Перевага онлайн-курсу полягає в тому, що він може охопити тисячі користувачів одночасно в будь-якій точці світу, якщо вони мають комп'ютер, планшет або смартфон і підключення до Інтернету. Ось чому, особливо після пандемії Covid-19, школи, університети, академії та навчальні заклади збільшили

використання електронного навчання. Але в цих ситуаціях утримати увагу учнів, які замість учителя опиняються перед екраном, нелегко. Щоб гарантувати успіх онлайн-курсу, важливо провести дослідження взаємодії між користувачами та цифровим пристроєм, щоб зробити використання платформи, яка передає курс, простішим і оперативнішим. Однією з характеристик успішного курсу електронного навчання є саме інтерактивність, оскільки вона дозволяє студенту активно діяти та застосовувати на практиці практичні поняття, засвоєні до того моменту, стикаючись із можливими реальними сценаріями роботи. Враховуючи ці передумови, у розробці курсу електронного навчання частина дизайну взаємодії відіграє фундаментальну роль, яка полягає у вивченні потреб користувача, щоб дозволити йому/їй легко взаємодіяти з платформою.

Що таке дизайн взаємодії

Згідно з Interaction Design Foundation, інтерактивний дизайн означає «дизайн продуктів і послуг», де фокус зосереджується не лише на елементі чи платформі, які потрібно розробити, а й на тому, як користувачі з ними взаємодіють. Це орієнтований на користувача підхід, який знаходиться в центрі дизайну. Основна мета дизайну взаємодії полягає у створенні продуктів, які дозволяють користувачам досягати своїх цілей легко та найкращим способом, сприяючи взаємодії між людиною та машиною (або цифровою платформою) і роблячи пропонувані послуги та системи більш доступними. Таким чином, перед дизайнером стоїть завдання зробити онлайн-сервіси та платформи легкими для використання користувачами, для яких вони були розроблені та створені [12].

Щоб досягти цих цілей, необхідно вивчити потреби, труднощі та цілі користувачів, щоб дизайн можна було налаштувати відповідно до потреб тих, хто має намір його використовувати. На основі дизайну взаємодії можна визначити чотири різні основні види діяльності, які становлять фундаментальні кроки для проектування взаємодії між користувачем і продуктом [14].

Зрозуміти, хто є передбачуваними користувачами, і визначити їхні потреби, щоб надати належну підтримку;

Розробка проектних пропозицій , які відповідають потребам і користувачам, визначеним вище;

Створюйте інтерактивні моделі продукту, який буде запропоновано, дозволяючи людям взаємодіяти з ними та оцінювати їх. Тому необхідно розробити інтерактивні версії проектних пропозицій, використовуючи прототипи;

Оцініть прийнятність пропозицій.

Дизайн взаємодії складається з п'яти вимірів , які представляють аспекти, які дизайнер взаємодії повинен враховувати під час розробки продукту. Вони є:

Слова : це 1D-вимір, який включає текст, написи кнопок, які дозволяють користувачеві отримати доступ до необхідної інформації. Текст, щоб забезпечити ефективну взаємодію, повинен бути простим і зрозумілим, надавати необхідну інформацію, не перевантажуючи користувача.

Візуальні представлення , тобто двовимірний вимір, який включає всі графічні елементи, такі як зображення та значки. Зазвичай вони доповнюють слова [11].

Фізичні об'єкти , тривимірний вимір простору, тобто засоби, за допомогою яких користувачі можуть взаємодіяти з продуктом. Це, наприклад, комп'ютерна миша.

Час (4D), який стосується відео , анімації та звуків, які змінюються з часом.

Поведінка (5D), яка стосується способу, у який попередні чотири виміри дозволяють користувачеві взаємодіяти з продуктом і як сервіс реагує на введення користувача.

Дизайн взаємодії є складовою ширшого дизайну досвіду , оскільки він зосереджується на використанні продукту та покращенні інтерактивного досвіду користувача. Якщо користувачу служби заважають сповіщення, тривалі анімації або відсутність чіткості, концентрація уваги буде зменшена, і продукт не зможе відповідати на потреби користувача.

Однією з основних сфер діяльності дизайнера взаємодії є розробка інтерфейсу продукту, який має бути привабливим, щоб привернути й утримати

увагу користувача. Як уточнює Usability.gov , для розробки взаємодії можна виконати шість кроків:

Визначати взаємодію користувача з інтерфейсом. Для цього може бути корисним запитати, що користувач може робити за допомогою миші (або пальця у випадку сенсорного екрана) і які команди він або вона може давати.

Надайте користувачам підказки щодо можливої поведінки , яку слід застосувати, звертаючи увагу на зовнішній вигляд інтерфейсу (колір, форму та розмір) та інформацію про те, що станеться, якщо взаємодія продовжиться.

Передбачте та усуньте помилки, підготувавши повідомлення про помилки, які дозволять користувачеві зрозуміти проблему та нададуть конкретні рішення для виправлення.

Зверніть увагу на час відгуку системи : реакція між дією та часом відгуку продукту має бути швидкою, а користувач має отримувати відповідний зворотний зв'язок;

Подумайте про кожен елемент інтерфейсу , намагаючись побачити, чи всі вони мають ефективний вимір взаємодії, зробивши їх достатньо великими, щоб користувач міг вибрати. Межі та кути також важливі, оскільки їх зазвичай можна використовувати для розміщення інтерактивних елементів, таких як спадні меню.

Важливість дизайну взаємодії в електронному навчанні

Вивчаючи обмеження, труднощі та переваги користувачів, можна вжити заходів для підвищення рівня задоволеності користувачів під час взаємодії з даним програмним забезпеченням. Цей механізм є фундаментальним у контексті електронного навчання , оскільки він дозволяє покращити зручність використання онлайн-курсу та зберегти увагу студентів, надаючи їм інструменти, необхідні для проходження всіх уроків. Щоб досягти цього, продукт повинен гарантувати адекватну доступність і просту, інтуїтивно зрозумілу та ефективну взаємодію. Крім того, навігація продуктом має бути швидкою та безперешкодною, дозволяючи студенту зосередитися на вмісті. У цій панорамі інтерфейс відіграє фундаментальну роль, оскільки разом зі своїми

компонентами він може привернути увагу учня та стимулювати навчальний процес. З цієї причини дизайн взаємодії також набуває першочергового значення при розробці курсів електронного навчання.

Слід враховувати три моменти:

Залучення : розробка привабливого інтерфейсу приваблює учнів, а використання кількох зрозумілих і простих елементів полегшує участь у навчальному процесі. Тому в електронному навчанні краще виключити непотрібну рекламу, посилання та модулі, які б лише відволікали увагу користувачів. Щоб привернути увагу, головна сторінка дуже важлива. Він містить назву курсу, його опис, огляд усіх уроків і запрошення до реєстрації.

Взаємодія : дизайн курсу повинен спонукати користувачів до взаємодії з ним. Для цього вирішальними є простота використання, інновації та функціональність платформи. Якщо учень може легко взаємодіяти з продуктом, він/вона буде більш мотивований продовжувати уроки, і його/її інтерес до курсу не втрачатиметься з часом.

Задоволення : коли студент задоволений своїм досвідом, він або вона намагатиметься повторити його та, урок за уроком, завершити курс електронного навчання. У розробці онлайн-курсу навігація відіграє вирішальну роль: швидка, проста та інтуїтивно зрозуміла навігація покращує досвід учня, роблячи його більш приємним у використанні.

Загалом, для того, щоб зробити користувацький досвід задовільним і утримати увагу учня, швидкість навігації, легкість взаємодії, приваблива графіка та ясність інформації мають вирішальне значення.

В результаті дослідження даного розділу я зрозумів для себе важливість інтерактивного дизайну для успіху освітніх додатків. Інтерактивність, адаптація до потреб користувачів, використання різних вимірів дизайну та забезпечення простоти і швидкості навігації є ключовими елементами, що забезпечують ефективне та приємне навчання. Для досягнення успіху в розробці освітніх платформ необхідно приділяти особливу увагу дизайну взаємодії, що дозволяє користувачам легко та з задоволенням взаємодіяти з продуктом.

1.5. Аналіз існуючих рішень

У iOS App Store і Play Store є різні типи навчальних програм, залежно від теми, яка цікавить кожного. Від вивчення нових мов, удосконалення математики чи навичок читання й до керування часом або завданнями студенти можуть вільно вибрати будь-яку програму, яку вони вважають за потрібну для своїх цілей.

QuestionPro став одним із найбільш використовуваних програм для опитування на світовому ринку. З функціональними можливостями, які найкращі дослідники хвалять за використання на його інтуїтивно зрозумілій платформі, простий і готовий до використання для немаркетологів. За допомогою LivePolls ви можете створювати динамічні опитування аудиторії в реальному часі, щоб заохотити участь у класі. Використання зображень і таймер для відповідей може викликати конкурентоспроможність у темі, що викладається в класі.

Додаток Green owl Duolingo продемонстрував, що вивчити мову не так складно, як усі думають. Завдяки більш ніж 40 доступним мовам ця програма пропонує спосіб вивчити або вдосконалити мову за допомогою занять тривалістю від 20 хвилин до інтенсивних занять по 1 годину на день.

Evernote — це вражаюча організаційна програма, розроблена для створення нотаток, планування розкладів, фотографування та створення рукописних нотаток, і все це в одній програмі. Однією з найкращих функцій цієї програми є синхронізація між усіма вашими пристроями.

Canva — це потужний інструмент дизайну, який можна використовувати в освіті для створення проектів, які не лише чудово виглядають, але й допомагають навчати студентів основам цифрового дизайну. Це безкоштовний інструмент, який дозволяє студентам і викладачам працювати з редагуванням фотографій, макетом дизайну тощо на простій у використанні платформі

Безкоштовна версія Wolfram Alpha дозволяє студентам бачити відповіді на складні математичні рівняння. У платних версіях Wolfram|Alpha студенти можуть бачити покрокові розв'язки завдань з математики, фізики та хімії.

За допомогою Google Docs ви можете створювати та редагувати текстові документи прямо у своєму веб-браузері, не потребуючи спеціального програмного забезпечення. Ще краще, кілька людей можуть працювати одночасно, ви можете бачити зміни, які люди вносять, і кожна зміна зберігається автоматично.

Genially — це веб-інструмент, доступний у безкоштовній версії, який дозволяє створювати анімовану інфографіку, інтерактивні презентації та навіть ігри-квести. Використовуйте цей інструмент, щоб цікаво представити вміст своїм студентам. Або запропонуйте його своїм студентам як наочний посібник під час їхніх усних доповідей.

Додаток Kids Academy є однією з найінноваційніших і найцікавіших програм для молодих учнів К-12. Їхня унікальна програма «навчання через гру» поєднує в собі найпередовіші адаптивні технології, штучний інтелект, персоналізоване навчання та гейміфікацію, щоб допомогти дітям зробити все можливе та отримати освіту, на яку вони заслуговують.

Ще один чудовий додаток К-12. «Навчи свого монстра читати» — відзначений нагородами навчальний додаток для дітей, який навчає дітей читати у веселій та спрощеній формі. Додаток охоплює перші два роки читання, від підбору букв і звуків до насолоди книгами. Додаток розроблено у співпраці з науковцями Рогемптонського університету та доповнює фази 2-5 затверджених Урядом Великобританії літер і звуків та інших основних систематичних синтетичних звукових програм.

Одна з найбільших переваг Google Classroom полягає в тому, що він простий у використанні та сприяє співпраці між учнями та вчителями. Вчителі можуть створити клас і створити список освітніх програм, наприклад завдань, кількома клацаннями миші. Вони можуть додавати учнів за іменем або надіслати їм код для приєднання. Після цього учні зможуть побачити, які завдання

потрібно виконати, взяти участь у дискусійних форумах або надіслати повідомлення викладачеві (приватно або через груповий чат).

Освітня індустрія пережила вибух технологій раптово після глобальної епідемії. Вся галузь відкриває нові зміни разом із технологіями. Справа не в тому, що технологія нещодавно з'явилася в освітньому секторі, вона існувала, але використання навчальних програм було обмеженим. Це була альтернатива, а тепер стала необхідністю. Це прискорило використання розробки освітнього програмного забезпечення за допомогою мобільних додатків і допомогло компаніям досягти нових висот, особливо в цьому освітньому секторі. Технологічні інструменти та платформи, які дозволяли вчителям спілкуватися зі своїми учнями, відстежувати їхній прогрес у навчанні або ділитися своїми курсами, очевидно, користувалися великим попитом під час дистанційного навчання.

Безперечно, освіта та навчання за допомогою технологій зробили процес швидшим, зрозумілішим і зручнішим. Багато навчальних закладів зробили для своїх викладачів і студентів необхідністю найкраще використовувати програми. Це призвело до того, що багато освітніх програм стали першим вибором і зробили різницю в освітньому секторі. Зі збільшенням використання платформ електронного навчання заклади можуть отримати такі переваги, як нижча вартість операцій, вища ефективність, легше управління та багато інших переваг для бізнесу. Мало хто з вас знає, хто вважає, що освітній сектор є третьою за популярністю категорією в розробці додатків. Це змушує нас думати, що освітні технології, безперечно, потужні та корисні для дітей, студентів, викладачів та інших зацікавлених сторін тієї ж галузі.

Багато шкіл, районів та установ прийняли ідею електронного та мобільного навчання. Вони надають своїм учням доступ до мобільних телефонів і портативних комп'ютерів і закликають їх використовувати їх з метою навчання. Хоча мобільне навчання може бути приємним для дітей, але якщо воно використовується правильно, воно також може бути неймовірно цінним для учнів будь-якого віку та навіть дорослих.

Незрівнянний аспект мобільного навчання зробив так багато глобальних ресурсів доступними онлайн. Отже, якщо ви не отримujete те, що хочете, з однієї платформи, ви можете шукати це десь. Шукайте іншу платформу. Навіть якщо ви користуєтесь додатком і чогось не розумієте, ви можете легко надіслати відгук і обговорити свою проблему чи пропозицію з розробником додатка чи фірмою, яка її виробляє.

Концепція електронного навчання вже зрозуміла для нас. Це чітко вказує на те, що такий спосіб навчання дозволить вам отримати доступ до інформації з будь-якого місця та в будь-який час. З доступом мобільних телефонів або смарт-пристроїв можливість завантажувати будь-який додаток стає зручною. Цей тип програмного забезпечення забезпечує гнучкість у двох аспектах: по-перше, воно дозволяє впроваджувати навчальні програми незалежно від місця розташування викладачів чи студентів. Другий спосіб полягає в тому, що використання хмарних сховищ дозволяє легко отримати доступ із багатьох пристроїв.

Ви можете скористатися всіма функціями освітнього програмного забезпечення на своєму телефоні та отримати від нього переваги.

Програми дозволять вам спілкуватися з будь-ким у всьому світі та навчатися у них найкращого. Ви можете отримати доступ до цього об'єкта з будь-якого місця, де є Інтернет. Насправді існує кілька програм, які також дозволяють працювати в режимі офлайн.

Порівняно з попередніми роками використання інформаційних технологій було зосереджено на фіксованих пристроях і для фіксованого використання. Однак на цьому рівні, коли технологічне створення та зростання процвітають, вони вимагають постійного доступу до бази знань. Мобільні пристрої часто знаходяться і належать одній людині, вони роблять навчальний процес безперервним. На відміну від традиційних методів навчання, учні можуть завершити роботу в будь-який зручний для них час, а вчителі можуть перенести пасивну частку навчання за межі аудиторії. Крім того, мобільні телефони завжди доступні зі значними перевагами, які дозволяють мати підручні матеріали, соціальну взаємодію та зручність використання, якщо у вас є

підключення до Інтернету. Мобільні додатки дозволяють продовжувати навчання в районах бойових дій і навіть зон катастрофи, демонструючи яскраве втілення спадкоємності.

Безумовно, діти люблять користуватися навчальними програмами. Вони заряджають енергією та розважають. Вчителям може бути важко переконати учнів бути уважними на уроці, але вони рідко намагаються змусити їх використовувати комп'ютер. Заняття фізкультурою проходять легко, тому що можна вийти на місце учня і зрозуміти його. На таких заняттях все, що вам потрібно зробити, це зробити заняття більш цікавими, що спонукатиме студентів автоматично зацікавитися заняттям. Як правило, ці заняття також є розважальними. Але є студенти, які не завжди можуть бути залучені до лекції, вони можуть зацікавитися певним предметом через додаток.

Використання освітніх програм може допомогти учням більше зацікавитися предметами, яких вони могли б уникати. Але загалом сама програма настільки приваблива завдяки інтерактивним зображенням, розповідям і рpts, відео, які залучають користувачів до вашої програми.

Насправді ми знаємо, що в Інтернеті є багато вмісту. Так само доступні варіанти для користувачів, щоб вибрати з їх улюблених платформ. Оскільки існує велика різноманітність, людям досить легко отримати до неї доступ. Вони можуть вибрати, що завгодно. Вони також можуть вплинути на велику кількість людей з усього світу, щоб отримати доступ до нього для різних тем або питань.

Завдяки наявності широкого спектру контенту у вас є шанс виділитися з натовпу. Ваш вміст може бути унікальним, а додаток – інноваційним. Це можна зробити лише за допомогою найновіших фреймворків, мови програмування, яка найкраще підходить для вашої програми та повністю відображає розроблену програму.

Однією з головних переваг освітніх програм є налаштування. Персоналізація є одним із найбільш інтерактивних методів навчання, що мотивує студентів все більше і більше підключатися до програми. Інтерактивні програми не змушують вас підключатися; натомість сеанси настільки інтерактивні та

спільні, що ви можете відчувати потребу в цьому пізніше. Освітні програми швидко стають кращим методом навчання для студентів, оскільки вони дозволяють їм навчатися з власною швидкістю та в комфорті власного дому.

Інтерактивне навчання також існувало в класах, але з віртуальним навчанням цей процес покращився ще більшою мірою. Крім того, перейдіть до наступного рівня складності на основі особистих уподобань і отриманої інформації, самостійно оцініть свої досягнення та виконайте додаткові дії, щоб закріпити вміст. Крім того, мобільний телефон дозволяє кожному студенту вивчати інформацію в найбільш зручному для нього форматі. Ви знайдете всі найкращі способи навчання за допомогою цих платформ електронного навчання, таких як РРТ, відео, gif, розповідь і багато інших інтерактивних способів.

Користувачі цієї програми можуть використовувати її в будь-який час. На відміну від традиційних шкіл, якщо учень пропустив період, йому чи їй доведеться робити нотатки в когось іншого. Але на онлайн-класах їм не потрібно хвилюватися, якщо вони пропустять онлайн-класи. Оскільки записані лекції доступні, користувачі можуть повертатися та відтворювати будь-яку лекцію, яку викладали, коли забажають.

Фактично, вони також мають можливість завантажити лекцію та посилатися на неї знову в майбутньому. Це дозволяє студентам навчатися в невимушеній обстановці без обмежень у часі. Вони завжди можуть знайти спосіб врегулювати свою роботу, життя та навчання, навіть якщо не хочуть цього.

Фактор вартості є однією з головних причин, чому бізнес виходить в Інтернет. Сьогодні про це говорять у всьому місті, дуже мало компаній насправді мають фізичні заняття, усі перейшли в Інтернет. Це означає бізнес-стратегію, яка демонструє грошові вигоди з плідними результатами. Отже, коли ви вирішите вибрати освітню програму, будьте впевнені, що програма пропонує економічно ефективні результати та прискорить загальну ефективність бізнесу.

Мобільні навчальні програми більш екологічні, ніж традиційні методи навчання, які використовують різні види паперу, олівці та ручки.

Інтелектуальне та систематичне навчання підтримуються освітніми програмами. Вся інформація в освітньому додатку добре організована, щоб стимулювати бажання студентів вчитися більше, і це робиться методичним чином.

Висококласне освітнє програмне забезпечення дозволяє учням без особливих зусиль слідувати логічному та плавному процесу. Крім того, ці навчальні програми дозволяють споживачам легко завантажувати довідкові конспекти. Крім того, мобільні програми пропонують швидші та ефективніші способи навчання.

Переїзд в Інтернет також має деякі недоліки. Технічні характеристики додатків відомі лише інженерам або розробникам освітнього програмного забезпечення. Саме вони можуть вирішити цю проблему. Програмне забезпечення — це програма, яка виконується на пристрої відповідно до інструкцій, закодованих у програмному забезпеченні під час кодування. Навіть якщо термін служби програмного забезпечення здається безперебійним, існують інші зовнішні змінні, які перешкоджають його безперебійному життю.

Іноді оновлення може спричинити зміни в додатку, зміну пристрою або будь-які інші фактори. Деякі зовнішні чинники також можуть впливати на розвиток ІТ. Нижче наведено деякі з проблем, які перешкоджають роботі програмного забезпечення, як-от проблеми сумісності програмного забезпечення, неможливість оновлення до нової версії та часті збої системи. Ось деякі з проблем, які спричиняють уразливість програмного забезпечення. Таким чином, це також порушує плавний досвід мобільного навчання.

Іншим фактором, що впливає на це, є апаратне забезпечення, підключене до програми навчання. Пов'язаний означає взаємопов'язаний якимись іншими засобами. Апаратне забезпечення, на відміну від програмного забезпечення, залежить від фізичних пристроїв. Фізичні гаджети можуть з часом зношуватися. Ці пристрої також можуть зіпсуватися через надмірне використання, пил і жорстке поводження з пристроєм. Окрім цього, можуть бути й інші фактори, які

можуть спричинити їх знос. Недоліком тут є те, що ви не зможете визначити, що пішло не так, і деякі елементи можуть заважати безперебійній роботі.

У суспільстві спостерігається зростаюче відчуження, а зміна уподобань щодо технологій і соціальних медіа призвела до більших проблем із психічним та емоційним здоров'ям. Отже, для людей, які вважають, що віртуальна взаємодія не надто ефективна, це може бути некорисним. Однак освітні додатки зменшують реальний світ і соціальні зв'язки, що викликає дискомфорт.

Зворотний зв'язок є головним мотиватором розвитку кожного учня. Навчання в онлайн-вчителя часто виявляється перешкодою для цього розвитку. Вчителі в цій ситуації надають зворотній зв'язок, але не мають достатньо часу, щоб розібратися з кожним учнем окремо та пояснити все детально. У результаті деякі студенти можуть відставати через прогалини у своїх знаннях, що призводить до невдалого завершення курсу.

Основною метою програмного забезпечення було вирішення проблем багатьох людей за допомогою однієї програми. Але це призвело до негативних наслідків, оскільки сотні людей доступні в Інтернеті, і вчителям важко відразу відповісти на проблеми всіх учнів. Для людини неможливо повернутися з усіма запитами відразу, що опосередковано вказує на відсутність належної реакції на проблеми. Студентам завжди потрібна пряма відповідь на їхню проблему, інакше з часом проблема зникає. Отже, коли програма не може її вирішити, вона стає некорисною.

Якщо немає технічного хисту, це завжди призведе до меншого використання технологій. Технологічні знання не однакові для всіх. Діти, які зручніше користуються планшетами та комп'ютерами, мають перевагу над тими, хто цього не робить. Насправді це також стосується батьків, якщо вони будуть знати про це менше, їхні діти постраждають.

Загалом діти з вищого соціально-економічного становища матимуть більший доступ до цифрового обладнання та будуть більш компетентними у його використанні. Ця прогалина породжує думку про те, що освітні технології сприяють нерівності у навчанні.

Навчання стало приємнішим і зручнішим завдяки використанню технологій. Оскільки студенти тепер мають доступ до нової платформи, а також можливість навчатися у відомих інструкторів з усього світу. Можна також назвати це дистанційною освітою.

Технічна підтримка є важливим фактором у передачі змісту навчальної програми користувачам. Створення сторінки технічної підтримки на веб-сайті курсу, надання всіх можливих проблем та шляхів їх вирішення в розділі поширених запитань (FAQ), надання онлайн-допомоги для вирішення проблем слухачів є критично важливими для ефективного онлайн-навчання. Надання можливості майбутнім слухачам ділитися знаннями та обмінюватися ідеями на онлайн-форумах, звертаючись за допомогою до викладача, коли це необхідно, зміцнить довіру до слухачів і підвищить їхню зацікавленість у курсі та його платформі в цілому. Дослідники вважають, що технічна підтримка має бути реактивною, тобто вирішувати проблеми слухачів без зайвих затримок. Учень повинен мати максимальну технічну підтримку, щоб користувацький інтерфейс був доступним (рис 2.1). Перед початком проходження онлайн-курсу студентам рекомендується здійснити віртуальну екскурсію середовищем курсу, щоб зрозуміти та вирішити технічні проблеми, якщо вони виникнуть.

Оскільки викладач фізично недоступний, щоб прояснити будь-які технічні сумніви, студенти будуть дуже розчаровані, коли зіткнуться з будь-якими технічними проблемами.

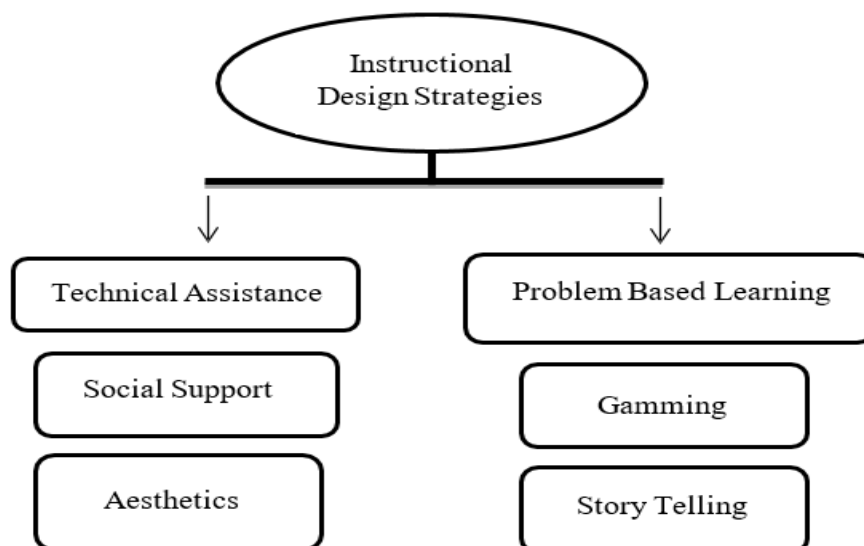


Рис. 1.1. Тематична карта для освітніх додатків.

Технічна підтримка є важливим фактором у передачі змісту навчальної програми користувачам. Створення сторінки технічної підтримки на веб-сайті курсу, надання всіх можливих проблем та шляхів їх вирішення в розділі поширених запитань (FAQ), надання онлайн-допомоги для вирішення проблем слухачів є критично важливими для ефективного онлайн-навчання. Надання можливості майбутнім слухачам ділитися знаннями та обмінюватися ідеями на онлайн-форумах, звертаючись за допомогою до викладача, коли це необхідно, зміцнить довіру до слухачів і підвищить їхню зацікавленість у курсі та його платформі в цілому. Дослідники вважають, що технічна підтримка має бути реактивною, тобто вирішувати проблеми слухачів без зайвих затримок. Учень повинен мати максимальну технічну підтримку, щоб користувацький інтерфейс був доступним. Перед початком проходження онлайн-курсу студентам рекомендується здійснити віртуальну екскурсію середовищем курсу, щоб зрозуміти та вирішити технічні проблеми, якщо вони виникнуть.

Оскільки викладач фізично недоступний, щоб прояснити будь-які технічні сумніви, студенти будуть дуже розчаровані, коли зіткнуться з будь-якими технічними проблемами. Вони дійсно не можуть продовжувати навчальний курс, і є всі можливості для повного розриву зв'язку.

Освітні додатки стали необхідністю після глобальної пандемії, прискоривши розвиток освітнього програмного забезпечення. Використання технологій зробило навчальний процес швидшим, зрозумілішим і зручнішим, дозволяючи навчальним закладам знизити вартість послуг, підвищити ефективність і покращити управління. Освітні технології виявилися потужними та корисними для учнів, викладачів та інших зацікавлених сторін.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ОСВІТНЬОГО ДОДАТКУ ДЛЯ ПЛАТФОРМИ АНДРОЇД

2.1. Аналіз сильних та слабких сторін існуючих додатків

Процес розробки освітнього додатка може бути складним і включати кілька етапів для створення навчального додатка, який буде цікавим, ефективним і зручним для користувача. Ось огляд ключових кроків, пов'язаних із розробкою освітніх програм:

Цей етап передбачає генерування ідей для розробки освітніх програм. Вам потрібно буде дослідити свою цільову аудиторію, визначити її потреби та проблемні точки, а також обдумати ідеї щодо того, як ваш додаток може задовольнити ці потреби.

Коли ви створили кілька ідей для своєї програми, настав час дослідити ринок і конкуренцію. Це допоможе вам зрозуміти поточний ландшафт розробки освітніх програм, визначити прогалини на ринку та вдосконалити концепцію вашої програми.

Етап проектування є критично важливим для створення зручного та цікавого освітнього додатка. Вам потрібно буде розробити інтерфейс користувача (UI), взаємодію з користувачем (UX) і візуальні елементи вашої програми. Цей етап включає каркасне моделювання, прототипування та створення керівництва по стилю.

Після завершення розробки настав час створювати освітню програму. Цей етап зазвичай включає програмування та кодування, а також інтеграцію таких функцій, як гейміфікація, соціальне навчання та інструменти оцінювання. Можливо, вам доведеться найняти команду розробників або аутсорсинг процесу розробки.

Розробка освітнього мобільного додатку включає важливий етап тестування додатка. Після створення програми її потрібно ретельно протестувати, щоб переконатися, що в ній немає помилок і вона працює

належним чином. Цей етап включає бета-тестування, тестування користувачами та перевірку якості.

Після перевірки та схвалення вашої освітньої програми настав час розгортати її в магазині додатків або на онлайн-ринку. Цей етап передбачає створення списку додатків у магазині, оптимізацію вашого додатка для пошукових систем і створення маркетингової стратегії для просування вашого додатка.

Створення навчального додатка може бути трудомістким і складним процесом, але за умови ретельного планування та виконання це може стати корисною працею, яка змінить наш спосіб навчання.

Розробка освітніх мобільних додатків відповідно до тенденцій ринку

Розробка навчальних програм є важливим аспектом розробки освітніх програм. Використовуючи найкращі методи розробки освітніх програм, ви можете створити освітню програму, яка буде цікавою, зручною та ефективною. Нижче наведено кілька практичних порад щодо розробки навчальних програм.

Дуже важливо розробляти освітню програму з урахуванням користувача. Розробляючи програму, ви повинні враховувати вік, стиль навчання та рівень освіти цільової аудиторії. Використовуйте відгуки бета-тестування та тестування користувачів, щоб вдосконалити дизайн програми.

Гейміфікація — це ефективний спосіб зробити навчання веселим і захоплюючим. Додавши ігрові функції, такі як нагороди, бали та значки, ви можете мотивувати користувачів навчатися та досягати прогресу в навчанні.

Адаптивне навчання — це підхід, який коригує процес навчання на основі прогресу та здібностей користувача. Впровадивши адаптивне навчання у свій додаток, ви можете персоналізувати процес навчання та надати користувачам вміст, який відповідає їхньому рівню знань.

Соціальне навчання — це підхід до спільного навчання, який передбачає групову діяльність, дискусійні форуми та зворотний зв'язок з однолітками. Включивши соціальне навчання у свій додаток, ви можете розвинути почуття спільноти та заохотити користувачів вчитися одне в одного.

Інструменти оцінювання – це ефективний спосіб оцінити прогрес користувача та надати відгук. Включивши інструменти оцінювання у свій додаток, ви можете надати користувачам уявлення про їхні сильні та слабкі сторони та скеровувати їх на шляху навчання.

Інтуїтивно зрозумілий інтерфейс користувача має вирішальне значення для створення зручної освітньої програми. Переконайтеся, що дизайн вашого додатка простий, простий у навігації та візуально привабливий.

Безперебійний користувацький досвід відіграє важливу роль у збереженні користувачів підключеними до навчальних програм. Переконайтеся, що функціональність вашої програми плавна та без збоїв, щоб користувачі могли легко переміщатися між різними розділами програми.

Популярні технології для розробки навчальних програм

Сфера розробки освітніх програм постійно розвивається, і розробники постійно досліджують нові технології, щоб створювати більш інноваційні та ефективні освітні програми. Ось кілька популярних технологій, які використовуються в розробці освітніх мобільних додатків:

Штучний інтелект (AI) і машинне навчання (ML) все частіше використовуються в розробці навчальних програм для створення персоналізованого навчального досвіду. Ці технології можуть аналізувати дані користувачів, щоб надавати персоналізований вміст і відгуки.

Доповнена реальність (AR) і віртуальна реальність (VR) — це захоплюючі технології, які можуть створювати захоплюючі навчальні враження. Ці технології можна використовувати для моделювання реальних ситуацій і покращення розуміння користувачем складних концепцій.

Технологію блокчейн можна використовувати для створення безпечних і прозорих цифрових записів про досягнення та прогрес учнів. Ця технологія також може сприяти створенню децентралізованих освітніх платформ.

Хмарні обчислення – це технологія, яка дозволяє програмам зберігати дані та виконувати складні обчислення на віддалених серверах. Цю технологію

можна використовувати для створення освітніх програм, які є масштабованими, безпечними та доступними з будь-якого місця [5].

Після огляду теоретичних відомостей перейдемо до створення прототипу застосунку.

PyCharm, доступний як кросплатформенний додаток, сумісний із платформами Linux, macOS і Windows. Витончено сидить серед найкращих IDE Python, PyCharm забезпечує підтримку версій Python 2 (2.7) і Python 3 (3.5 і вище).

PyCharm постачається з безліччю модулів, пакетів та інструментів, щоб пришвидшити розробку Python, одночасно скорочуючи зусилля, необхідні для виконання того ж самого в значній мірі. Крім того, PyCharm можна налаштувати відповідно до вимог розробки та особистих уподобань. Його вперше було опубліковано в лютому 2010 року. Окрім аналізу коду, PyCharm має такі функції:

- графічний налагоджувач;
- інтегрований блок-тестер;
- підтримка інтеграції систем контролю версій (VCS);
- підтримка науки про дані за допомогою Anaconda.

Основною причиною, по якій Pycharm створив цю IDE, було програмування на Python і робота на кількох платформах, таких як Windows, Linux і macOS. IDE містить інструменти аналізу коду, налагоджувач, інструменти тестування, а також параметри контролю версій. Він також допомагає розробникам створювати плагіни Python за допомогою різних доступних API. IDE дозволяє нам працювати з декількома базами даних безпосередньо без інтеграції з іншими інструментами. Хоча він спеціально розроблений для Python, файли HTML, CSS і Javascript також можна створювати за допомогою цієї IDE. Він також поставляється з прекрасним інтерфейсом користувача, який можна налаштувати відповідно до потреб за допомогою плагінів.

PyCharm — надзвичайно популярне середовище розробки Python. Інтегроване середовище розробки або IDE містить редактор коду та компілятор для написання та компіляції програм на одній або кількох мовах програмування.

Крім того, IDE поставляється з великою кількістю функцій, які полегшують комплексну розробку програмного забезпечення. Оскільки IDE призначає різні кольори різним об'єктам програмування, зазвичай відоме як підсвічування синтаксису, воно стає більш доступним для:

Розрізняти різні сутності програмування, такі як клас і функція, і виявляти їх.

Шукайте неправильні ключові слова.

Прочитайте та зрозумійте код.

Більшість IDE мають функцію автозаповнення, яка створює пропозиції під час написання коду. Це робить написання коду більш ефективним, швидким і менш схильним до помилок і опечаток. Інші стандартні функції, які пропонує сучасна IDE:

Вікно редактора проекту для ефективного керування та організації файлів, необхідних для програми/проекту.

Перевірка виводу коду, написаного за допомогою вікна виводу

Пропозиції щодо усунення помилок і попереджень

Низка модулів і пакетів, доступних в одному місці

Особливості PyCharm

1. Інтелектуальний редактор коду

PyCharm поставляється з розумним редактором коду, який полегшує написання високоякісного коду Python. Він пропонує покращений рівень розуміння та читабельності коду за допомогою різних кольорових схем для ключових слів, класів і функцій, тобто підсвічування синтаксису та помилок.

Окрім функції інтелектуального завершення коду, редактор коду генерує інструкції для завершення поточного коду. Виявлення помилок і проблем набагато зручніше разом із інтеграцією Linter і швидкими виправленнями.

2. Наявність інструментів інтеграції

PyCharm підтримує інтеграцію ряду інструментів. Ці інструменти варіюються від допомоги в підвищенні продуктивності коду до полегшення роботи з проектами науки про дані. Деякі з найважливіших інструментів інтеграції, доступних для PyCharm, включають:

Anaconda – безкоштовний дистрибутив Python із відкритим вихідним кодом, призначений для наукових обчислень із спрощеним керуванням пакунками та розгортанням.

IPython – надійна командна оболонка для інтерактивних обчислень.

Kite – плагін автозаповнення на основі штучного інтелекту.

PyLint - засіб перевірки вихідного коду, помилок і якості.

pytest — фреймворк для написання невеликих тестів для коду Python.

WakaTime – інформаційна панель розробника з показниками продуктивності та автоматичним відстеженням часу

3. Наука про дані та машинне навчання

PyCharm постачається з підтримкою наукових бібліотек, таких як Matplotlib і SciPy, щоб допомогти розробникам Python виконувати проекти з обробки даних і машинного навчання.

4. Google App Engine

Google App Engine або безпосередньо App Engine — це платформа PaaS і хмарних обчислень, призначена для розробки та розміщення веб-додатків. Він пропонує автоматичне масштабування для веб-додатків. Професійна версія PyCharm забезпечує підтримку Google App Engine.

5. Інтегроване налагодження та тестування

IDE поставляється з підтримкою налагодження та тестування програм. Щоб досягти того ж, PyCharm має інтегрований налагоджувач Python і інтегроване модульне тестування з охопленням коду рядок за рядком.

6. Мультитехнологічна розробка

Розробники Python також можуть використовувати PyCharm для створення веб-додатків. Таким чином, Python IDE забезпечує підтримку популярних веб-

технологій, включаючи CoffeeScript, CSS, HTML, JavaScript, TypeScript. Крім того, він також включає підтримку Cython, мов шаблонів і SQL.

Редагування в реальному часі також доступне в PyCharm, тобто розробники можуть створювати/змінювати веб-сторінку, одночасно надаючи її в режимі реального часу. Таким чином, за змінами можна стежити безпосередньо у веб-браузері. Також доступне створення веб-додатків за допомогою AngularJS або NodeJS.

7. Навігація по проекту та коду

Функція навігації кодом значно полегшує розробникам перехід до класу, функції або файлу. Це також допомагає значно скоротити зусилля та час, необхідні для редагування та вдосконалення коду Python. Перегляди структури файлів і спеціалізовані види проектів легко доступні.

Режим об'єктива дозволяє розробнику ретельно перевіряти та налагоджувати весь вихідний код Python. За допомогою навігації по коду місцезнаходження елемента, змінної тощо виконується практично миттєво. Розробники можуть швидко переходити між класами, файлами та методами.

8. Рефакторинг

Функція рефакторингу в PyCharm допомагає покращити внутрішню структуру програми Python, не впливаючи на її зовнішню продуктивність. Внесення змін до локальних і глобальних змінних є ефективним і швидким.

Метод екстракту також існує для розділення розширених класів і функцій. Інші корисні функції рефакторингу коду включають:

- ввести константу;
- введіть змінну;
- видалення;
- пермістити;
- перейменувати.

Також для зрозумілості логіки програми розробимо UML діаграми застосунку (рис 2.1-2.3)

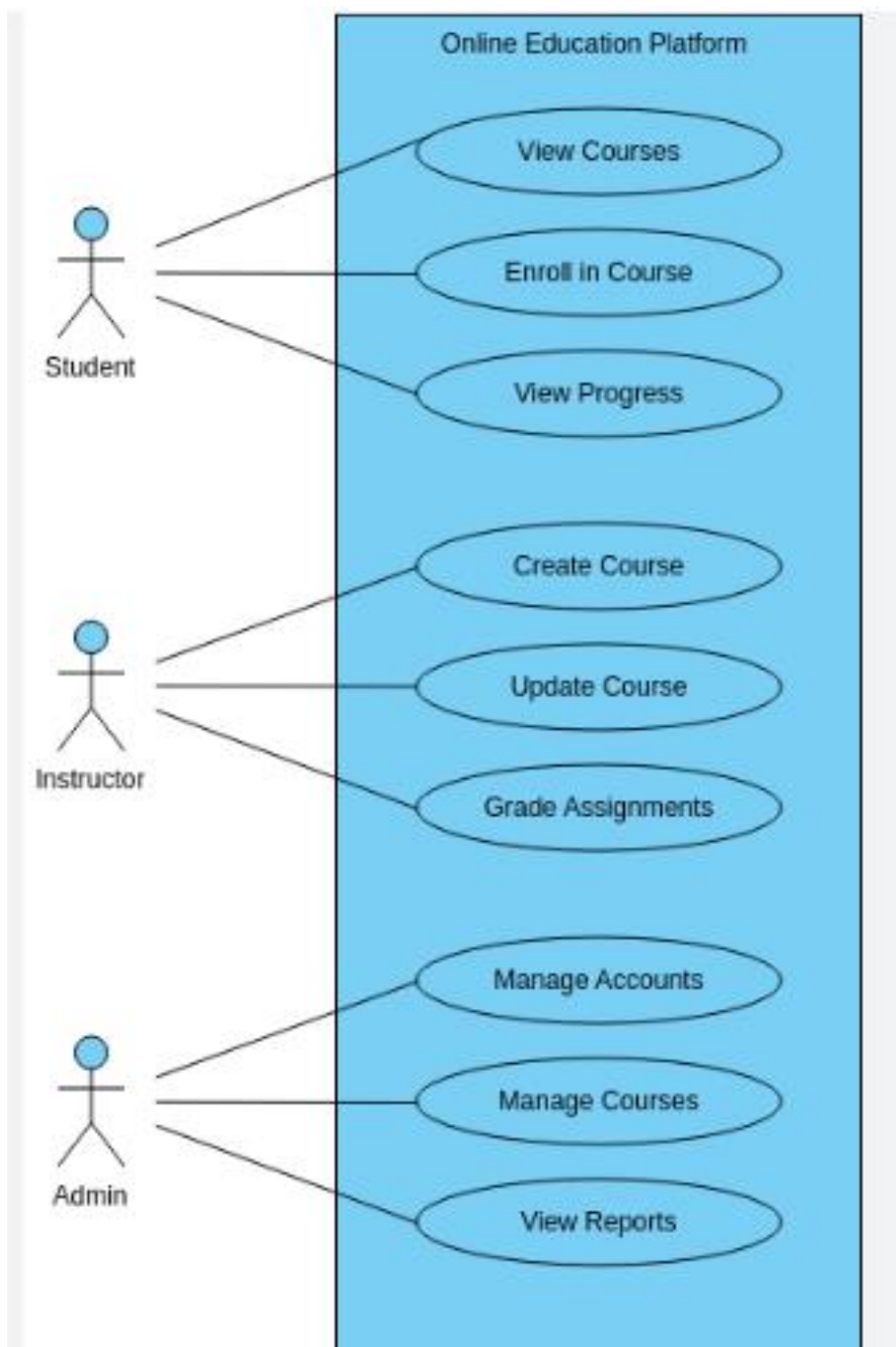


Рисунок 2.1. UML Use case застосунок.

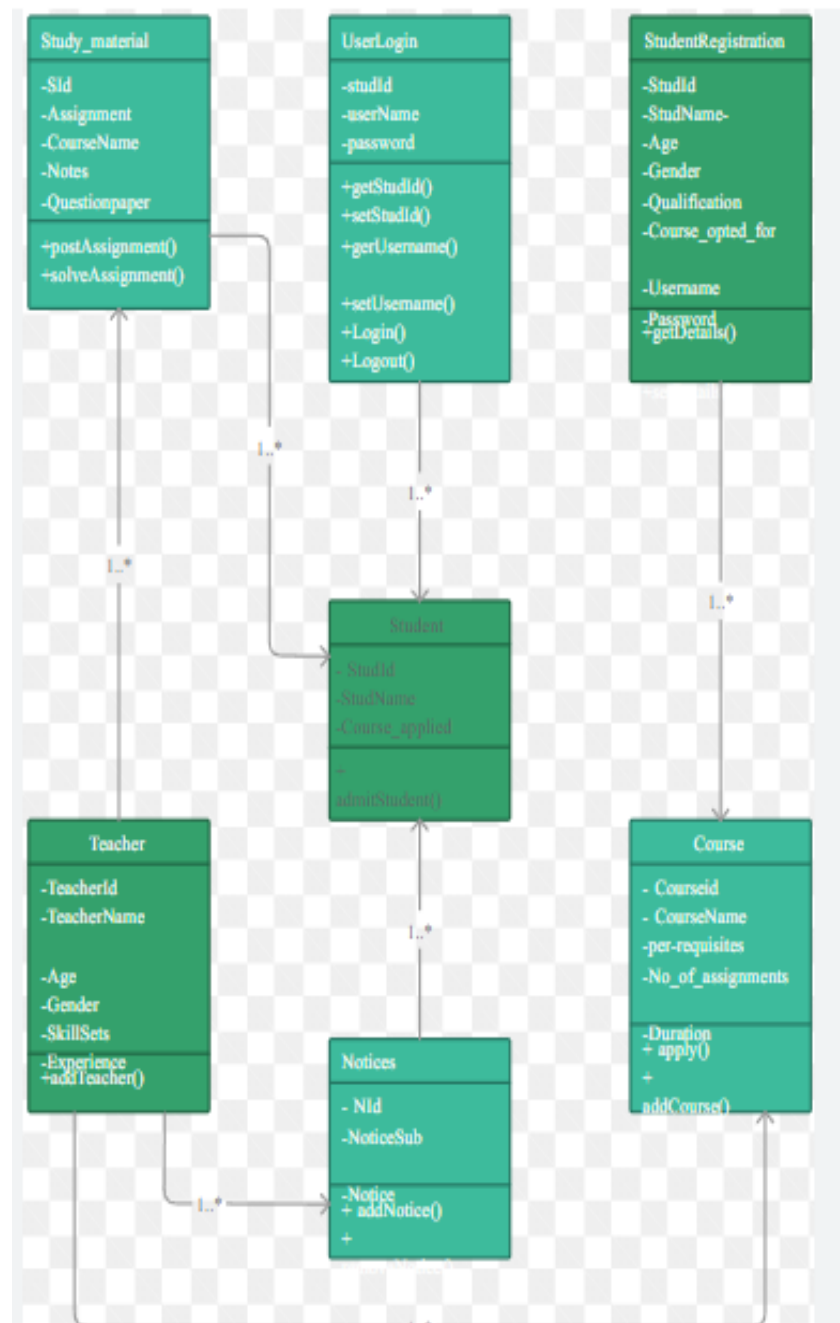


Рис. 2.2. UML діаграма класів застосунку.

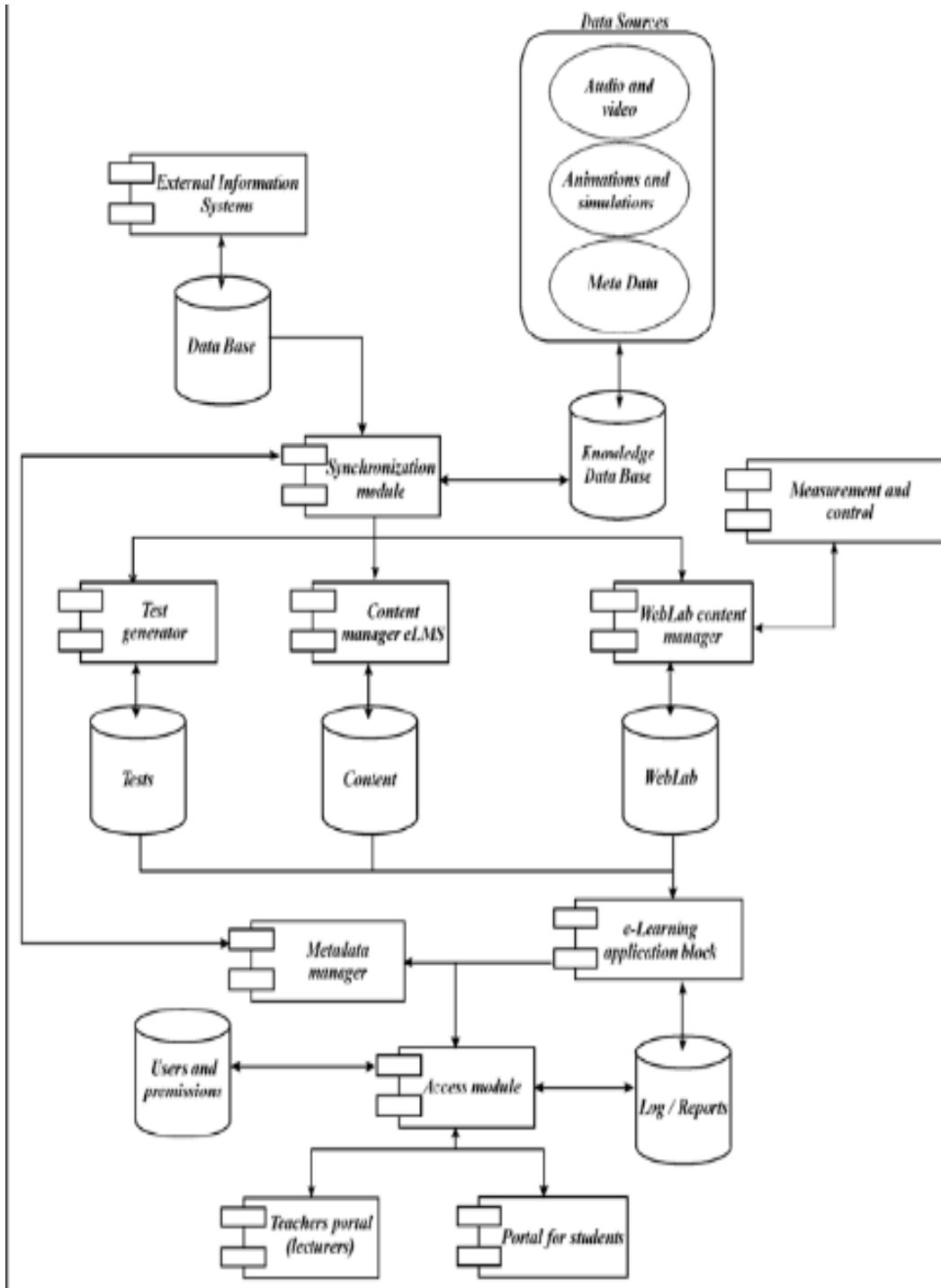


Рис. 2.3. UML діаграма класів.

```

import tkinter as tk
from tkinter import messagebox
import hashlib
import json

```

```

class RegistrationForm:
    def __init__(self, master):
        self.master = master
        master.title("Форма реєстрації")

        self.label_username = tk.Label(master, text="Логін:")
        self.label_username.grid(row=0, column=0, sticky=tk.E)

        self.label_password = tk.Label(master, text="Пароль:")
        self.label_password.grid(row=1, column=0, sticky=tk.E)

        self.entry_username = tk.Entry(master)
        self.entry_username.grid(row=0, column=1)

        self.entry_password = tk.Entry(master, show="*")
        self.entry_password.grid(row=1, column=1)

        self.submit_button = tk.Button(master, text="Зареєструватися",
command=self.register_user)
        self.submit_button.grid(row=2, column=0, columnspan=2)

    def register_user(self):
        username = self.entry_username.get()
        password = self.entry_password.get()

        if not username or not password:
            messagebox.showerror("Помилка", "Будь ласка, заповніть всі поля.")
            return

```

```

hashed_password = hashlib.sha256(password.encode()).hexdigest()

with open("users.json", "a+") as file:
    file.seek(0)
    users_data = file.read()
    if users_data:
        users = json.loads(users_data)
    else:
        users = {}
    if username in users:
        messagebox.showerror("Помилка", "Користувач з таким логіном
вже існує.")
        return
    users[username] = hashed_password
    file.seek(0)
    json.dump(users, file)

messagebox.showinfo("Інформація", "Користувач успішно
zareestrovaniy.")
self.master.destroy()

```

В даному фрагменті ми побудували для програми алгоритм реєстрації та авторизації користувача в застосунку.

З урахуванням аналізу сильних та слабких сторін існуючих додатків, можна зробити наступні висновки: Успішний процес розробки освітнього додатка вимагає ретельного планування та виконання кількох етапів, включаючи генерування ідей, аналіз ринку, проектування, розробку, тестування та розгортання.

Розробники освітніх додатків повинні враховувати потреби та характеристики цільової аудиторії, гейміфікацію, адаптивне навчання, соціальне навчання та інструменти оцінювання для створення ефективних програм. Популярні технології, такі як штучний інтелект, доповнена реальність, блокчейн та хмарні обчислення, можуть бути використані для створення інноваційних та ефективних освітніх програм.

Щодо конкретного випадку, PyCharm виявляється потужним інструментом для розробки програм, забезпечуючи розумний редактор коду, інтеграцію з різними інструментами та підтримку наукових бібліотек, що допомагає розробникам Python у створенні якісного програмного забезпечення. У разі використання PyCharm для створення освітнього додатка, розробники отримують значні переваги, включаючи зручність, продуктивність та доступ до різноманітних інструментів для створення високоякісних програм.

2.2. Розробка каркасів (Wireframes) та макетів (Mockups)

При запуску проекту з командою розробників вся проектна документація повинна бути укладена і підписана. Не завжди вдається розрізнити етапи та окремі елементи.

У цьому підрозділі будуть описані початкові етапи створення дизайну, створення каркаса, макета та прототипу. Ми окреслюємо розбіжності між ними, окремі випадки їх застосування та способи їх створення.

Це перший крок, основа, необхідна для початку створення будь-якого продукту. Wireframe будує основне представлення дизайну, що служить певним цілям. Це можна порівняти зі створенням конструктивного устрою будівлі та розрахунком напруженого навантаження.

Основне завдання при цьому – визначити змістовні елементи майбутнього інтерфейсу, загальний потік поведінки відвідувачів. Не потрібно звертати увагу на елементи чи кольори.

В основному його можна створити за допомогою ручки та аркуша паперу. Зробіть цей крок найменш трудомістким, але все ж важливим кроком з точки зору планування. Це також можна зробити за допомогою проектування програм.

Базовий каркас можна зробити за кілька годин або днів залежно від складності виробу.

Це дає можливість переглядати товар у макрозйомці та прискорює всі комунікаційні процеси.

Це статичне зображення дизайну майбутнього цифрового продукту. Макет включає графічні та функціональні елементи, які поєднані за каркасною схемою. Він містить фактичні кнопки, зображення та типографіку, а також графіку навігації.

Макет є найкращим етапом для обговорення візуалізації інтерфейсу, оскільки він приносить більше, ніж сірі лінії будь-якого каркасу. Його легше розпізнати людям, які не знайомі з дизайном чи розробкою.

Оскільки він містить різноманітні графічні елементи, це неможливо зробити без застосування спеціалізованого програмного забезпечення.

За допомогою макета можна побачити всі значущі елементи продукту, ввести необхідні корективи та додати деталі.

У порівнянні з використанням каркасу, коли ви бачите лише сірі лінії, макет виглядає більш яскравим. Візуальне представлення стає легше розпізнати та запропонувати необхідні зміни.

Прототип — це інтерактивна версія макета, яка показує взаємодію користувача на певному пристрої. Він відрізняється від кінцевого продукту тим, що він не підключений до внутрішніх баз даних.

На цьому етапі ви можете протестувати інтерфейс перед остаточним затвердженням і початком розробки. Це допоможе вам заощадити кошти, які ви могли б витратити на розробку зайвих екранів або інших елементів інтерфейсу.

З усіма взаємодіями та анімаціями користувач бачить продукт таким, яким він мав бути. Можна провести тести A/B і перевірити, чи відповідає продукт

початковим цілям. На цьому кроці можна знайти всі приховані проблеми та слабкі місця.

Маючи інтерактивний інтерфейс (рис 2.4), кожен може побачити, як працює система, і запропонувати деякі виправлення або, можливо, відсутні елементи чи екрани. Можна знайти кращий маршрут, ніж було запропоновано спочатку.

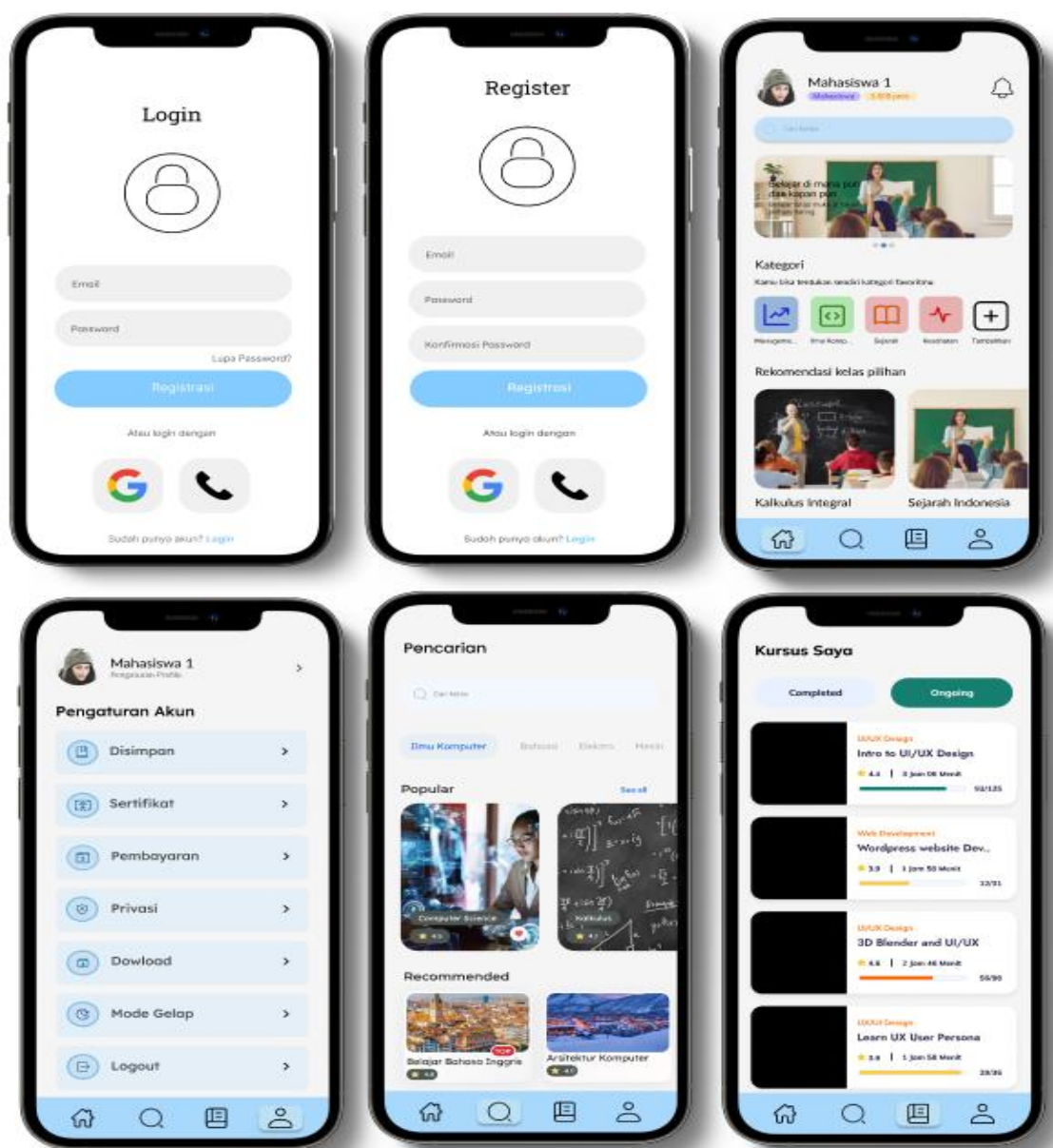


Рисунок 2.4. Розробка макету освітнього додатку.

Прототип може служити мінімальним продуктом, щоб продемонструвати можливим інвесторам, як має працювати кінцевий продукт. Вони можуть взяти

його на «тест-драйв» і перевірити, чи вартий він їхніх інвестицій. Прототип забезпечує кращий досвід, ніж каркас або макет, і він коштує менше, ніж уся розробка продукту.

Усі ці кроки мають одну мету, а саме створення ефективного дизайну, який буде реалізовано за допомогою коду. Вони є логічним продовженням один одного і мають свої важливі особливості для загального ходу розвитку. У результаті не йдеться про те, чи опустити деякі з них; це питання про те, як розумно їх запровадити та отримати максимальну віддачу від їх використання.

Розробка каркасів (Wireframes) та макетів (Mockups)" надає комплексний огляд ключових етапів процесу дизайну цифрових продуктів. Від початкового створення каркасів, які визначають структуру і функціональність, до більш деталізованих макетів, які відображають графічні та функціональні елементи, текст ілюструє важливість кожного етапу та їх взаємозв'язок.

2.3. Проектування інтерактивного дизайну

Дизайн взаємодії полягає в конвергенції взаємодії людини з комп'ютером. Його мета полягає в тому, щоб перетворити цифрові продукти з механічного та холодного досвіду на такі, які відчуються природно та особисто.

Іншими словами, дизайн взаємодії – це процес, який робить веб-сайти та програми інтерактивними.

Розробники взаємодії зосереджуються на тому, чого хочуть досягти користувачі, взаємодіючи з цифровим продуктом. Вони дають нам досвід, який відповідає нашим потребам і забезпечує наш успіх.

Дизайн взаємодії визначає ці точки взаємодії та визначає зворотний зв'язок, який ми отримуємо, коли виконуємо ці дії.

Інтерактивний дизайн зосереджується на тому, щоб це рух вперед-назад відчувався плавним і природним. Замість того, щоб комп'ютер керував нами

одностороннім обміном, ми стаємо частиною постійного діалогу, проходячи кожен крок.

Дизайнери взаємодії сприяють співпереживанню та зв'язку. Вони розуміють психологію та мотивацію цільової аудиторії та адаптують взаємодію, щоб ми відчували себе добре.

Запустити та запустити HoloViews легко. Він чудово працює з Python 3 на Linux, Windows або Mac, і це плавний оператор із Jupyter Notebook і JupyterLab.

```
# Встановити HoloViews і Bokeh за допомогою conda
```

```
conda install -c pyviz holoviews bokeh
```

Ця команда не тільки отримує HoloViews, але й захоплює звичайні пакети, які добре поєднуються з нею. Самому HoloViews просто потрібні Numpy, Pandas і Param, щоб зробити свою справу.

Якщо ви не використовуєте Conda, завантажте HoloViews з GitHub і встановіть його, виконавши такі дії.

```
# Клонувати HoloViews з GitHub
```

```
git clone git://github.com/holoviz/holoviews.git
```

```
# Перемістити до каталогу HoloViews
```

```
cd holoviews
```

```
# Встановити HoloViews
```

```
pip install -e .
```

Щоб отримати практичні навички роботи з HoloViews, давайте скористаємося наборами даних, пов'язаних із транспортом у Нью-Йорку. Ми почнемо з таблиці інформації про станції метро, завантаженої з файлу CSV:

```
# Імпортуйте необхідні бібліотеки,
```

```
імпортуйте pandas як pd # Для маніпуляції та аналізу даних
```

```
імпортуйте numpy як np # Для числових операцій
```

імпортуйте holoviews як hv # Для інтерактивної візуалізації даних
із holoviews імпортуйте опції # Для налаштування параметрів малювання
hv.extension('bokeh') # Активуйте Боке як розширення малювання

Як в результаті було створено оформлення просте та зрозуміле для базового користувача з функціоналом (рис 2.5-2.6):

- зареєструватися
- авторизуватися
- додати курс
- записатися на курс
- мій профіль

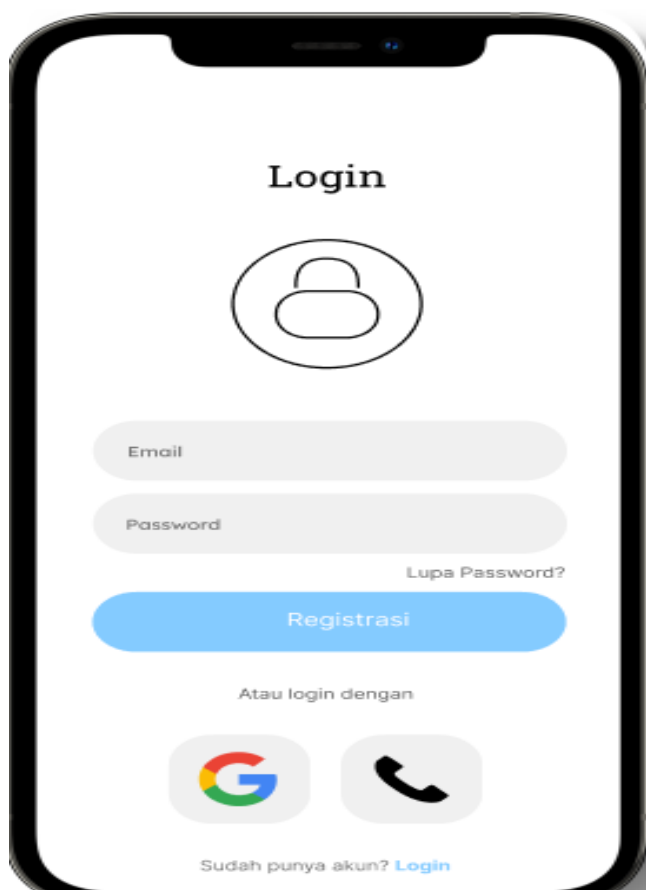


Рисунок 2.5. Форма авторизації.



Рисунок 2.6. Розробка форми взаємодії між користувачем та додатком.

РОЗДІЛ 3. ТЕСТУВАННЯ РОЗРОБЛЕНОГО ЗАСТОСУНКУ

3.1. Створення користувацьких сценаріїв (User Stories)

Усім відомо, що тестування програмного забезпечення та тестування мобільних пристроїв тісно пов'язані з керуванням продуктом загалом та історіями користувачів зокрема.

Ефективність контролю якості продукції страждає від непередуманих і непередуманих історій користувачів. Коли компанія з тестування програмного забезпечення суворо наполягає на використанні лише класичної схеми «як...», результати можуть бути неочікуваними. А іноді такий підхід просто не підходить до конкретної ситуації.

Найкраща ідея для такої ситуації — відмовитися від кліше та створити історію користувача по-своєму. Тестер повинен переконатися, що нова історія є короткою, чіткою та написаною у форматі, схожому на історію. Краще продемонструвати, що нова історія користувача корелює з існуючою функціональністю продукту.

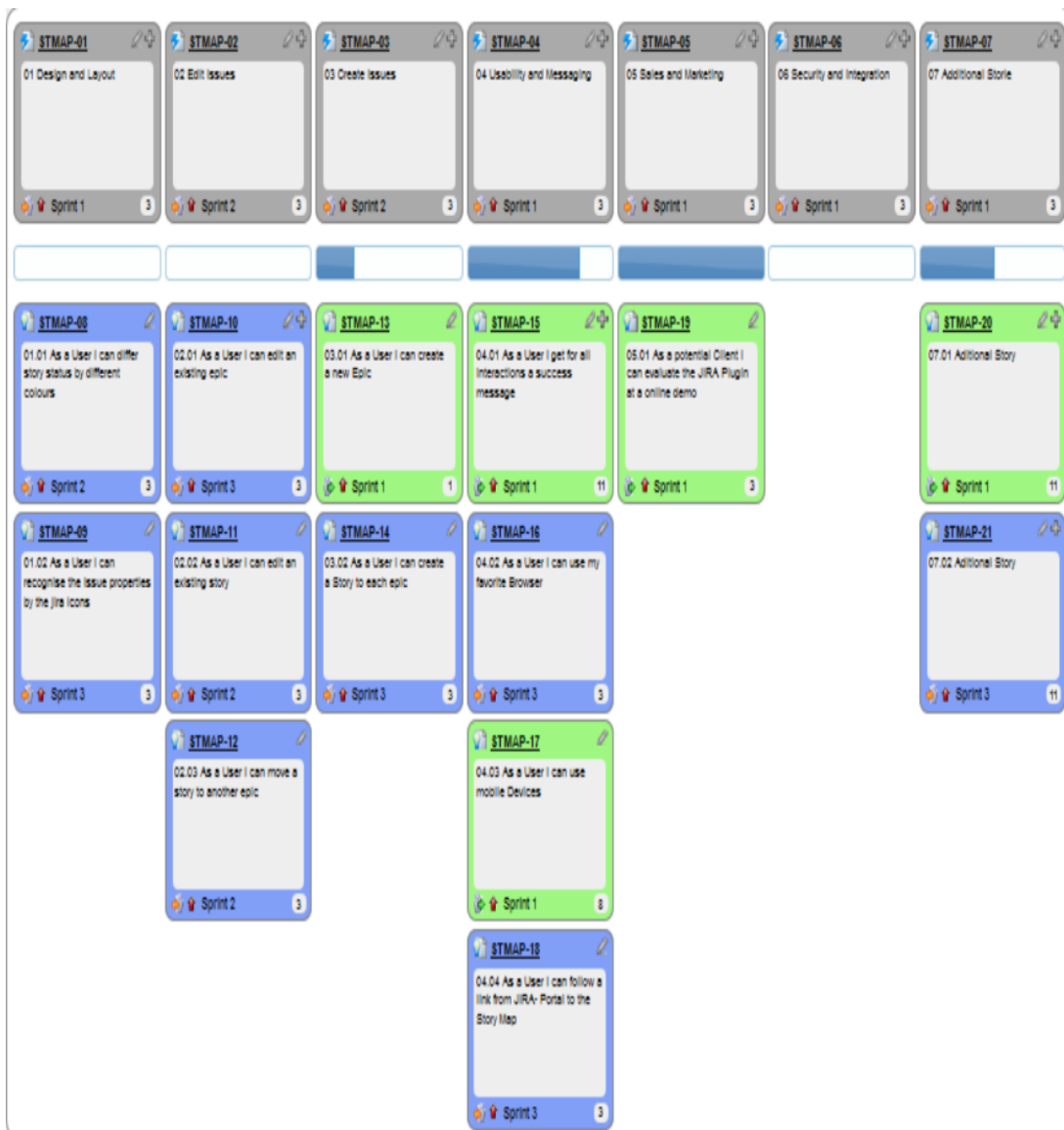


Рисунок 3.1. User Stories для освітнього проектного додатку.

Чітко пояснюється мотив історії користувача, додаються всі необхідні креслення, схеми та ескізи, надаються інформативні тести користувача. У цьому випадку шанс, що ця нова ідея буде прийнята і схвалена, дійсно великий.

Для правильного проведення всього вищесказаного розробники і тестувальники повинні бути проінформовані про основні технічні моменти

користувацької історії. Необхідно обговорити можливі винятки, які повинні бути прийняті до уваги технічними експертами. При дотриманні всіх перерахованих нюансів позитивний результат не змусить себе чекати.

Дуже важливо мати на увазі, що як ручне тестування, так і автоматичне тестування залежить від якості створених історій користувачів. Тестер не повинен нехтувати ретельною підготовкою складових процесу тестування.

3.2. Організація та проведення користувацького тестування

Керування мобільними пристроями (MDM) — це технологія, яка дозволяє школам і установам керувати та захищати мобільні пристрої, що використовуються в освітніх цілях. Цю технологію можна використовувати для забезпечення безпечного розгортання освітніх програм і керування ними.

Використовуючи ці популярні технології в розробці освітніх програм, розробники можуть створювати інноваційні та ефективні освітні програми, які відповідають постійним потребам учнів і викладачів. Йдучи далі, важливо розуміти той факт, що технологія різна для кожного типу освітнього мобільного додатка. Отже, технологію слід обирати залежно від типу освітньої програми.

Розробка освітніх програм: монетизація та маркетингові стратегії

Монетизація та маркетинг освітніх програм є важливим аспектом розробки навчальних програм. Ось кілька найкращих стратегій монетизації та маркетингу освітніх програм:

Модель Freemium передбачає надання безкоштовної версії програми з обмеженими можливостями та платної версії з додатковими функціями. Ця модель може допомогти збільшити базу користувачів програми та отримати дохід від платних підписок.

Покупки в програмі включають пропозицію користувачам придбати додатковий вміст, функції або віртуальні товари в програмі. Цю модель можна

використовувати для отримання прибутку від користувачів, які готові платити за додатковий вміст або функції.

Модель підписки передбачає надання користувачам доступу до преміум-вмісту або функцій за регулярну плату. Цю модель можна використовувати для отримання постійного прибутку та створення бази лояльних користувачів.

Реклама — поширена стратегія монетизації освітніх програм. Розробники можуть пропонувати рекламний простір у програмі для рекламодавців або використовувати рекламні мережі для показу націлених оголошень користувачам.

Маркетинг у соціальних мережах є потужним інструментом для просування освітніх програм. Розробники можуть використовувати платформи соціальних медіа, щоб створити спільноту навколо програми, взаємодіяти з користувачами та рекламувати програму ширшій аудиторії.

Після розробки переходимо до функціонального тестування додатку.

Спочатку перевіримо функцію реєстрації користувача (рис. 3.2)

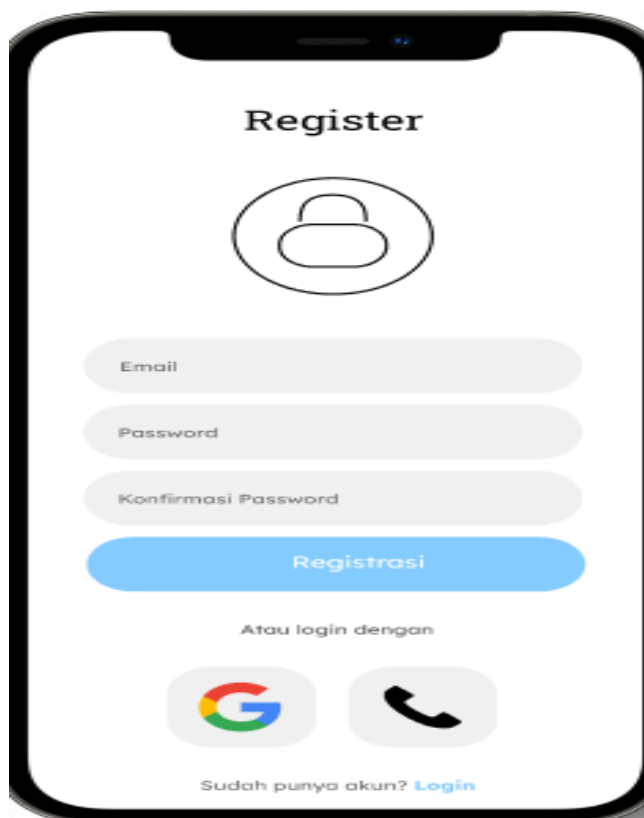


Рисунок 3.2. Реєстрація користувача.

Після по зареєстрованим даним перевіряємо функцію авторизації (рис. 3.3)

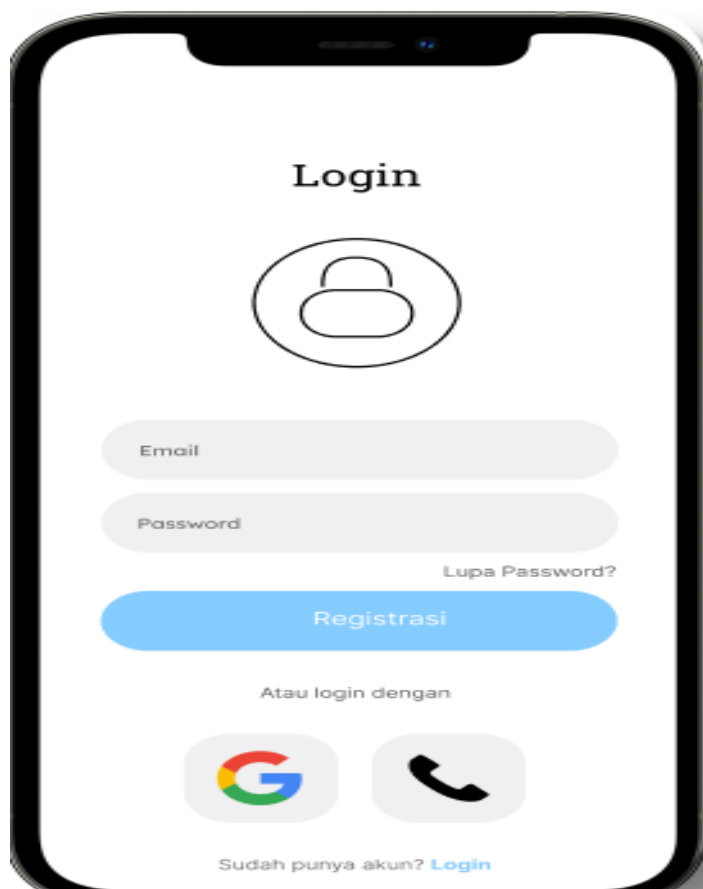


Рисунок 3.3. Авторизація користувача.

Як завершився процес авторизації користувач переходить на головне меню де може записатися на існуючий курс, додати свій курс, редагувати його (рис 3.4).



Рисунок 3.4. Меню взаємодії з курсами.

Також кожен користувач має можливість переглядати інформацію про себе та на які курси він записаний(рис 3.5)



Рисунок 3.5. Особистий кабінет користувача.

Оптимізація магазину додатків

Оптимізація магазину додатків (ASO) передбачає оптимізацію назви, опису, ключових слів і візуальних елементів програми для покращення її видимості в результатах пошуку в магазині додатків. Ця стратегія може допомогти збільшити видимість програми та залучити більше завантажень.

Інфлюенсер-маркетинг передбачає співпрацю з впливовими блогерами, особистостями в соціальних мережах або педагогами для просування програми серед своїх підписників. Ця стратегія може допомогти охопити ширшу аудиторію та підвищити довіру до програми.

3.3. Аналіз результатів тестування

Розробка освітніх додатків надала викладачам і учням інноваційні інструменти для підвищення ефективності навчання. Завдяки наявності різноманітних мобільних пристроїв навчання більше не обмежується аудиторіями та підручниками. Освітні програми дають учням свободу навчатися у власному темпі, у власний час і в комфорті власного середовища. Крім того, розробка навчальних програм зробила навчання більш захоплюючим та інтерактивним, полегшуючи розуміння складних концепцій.

Однак розробка освітнього додатка є складним і складним процесом. Це вимагає глибокого розуміння цільової аудиторії, освітніх цілей і останніх технологічних тенденцій. Розробники додатків повинні використовувати найкращі практики проектування, розробки, тестування та розгортання інтерфейсу користувача, щоб переконатися, що їхні додатки ефективні, надійні та безпечні.

Незважаючи на труднощі, пов'язані з розробкою навчальних програм, попит на освітні програми продовжує зростати. Оскільки технології продовжують розвиватися, освітні програми, безсумнівно, відіграватимуть дедалі важливішу роль у освітньому секторі. Тому вкрай важливо, щоб викладачі, розробники та інші зацікавлені сторони в освітньому секторі взяли на

озброєння розробку освітніх додатків і використали їх переваги, щоб змінити процес навчання на краще.

Як в результаті було створено застосунок для навчання простий та зрозумілий для базового користувача з функціоналом :

- зареєструватися
- авторизуватися
- додати курс
- записатися на курс
- мій профіль

Етапи тестування відобразимо в таблиці 3.1.

Таблиця 3.1.

Результати тестування.

Test case	Дії	Очікування	Статус
Перевірка авторизації	Авторизуватися по даним користувача	Якщо користувач є то система видасть повідомлення та нагадає що потрібно при потребі змінити пароль	Пройдено
Перевірка елементів	На головній сторінці перевірити чи добре відображаються елементи	Якщо не завантажується елемент то є піктограма хрестика	Пройдено
Перевірка запису	Користувач записується на курс	Після запису повинно з'явитися повідомлення про успішність запису на курс	Пройдено
Перевірка швидкості	Завантажити сторінку	Сторінка повинна швидко завантажитися	Пройдено

ВИСНОВКИ

У даній кваліфікаційній роботі була проведена розробка інтерактивного дизайну освітнього додатку Pyhrome на платформі Android.

У першому розділі роботи проведено аналіз існуючих освітніх додатків таких як:

QuestionPro

LivePolls

Green owl Duolingo

Evernote

Kids Academy

Canva

Wolfram Alpha

Та зроблено висновки, що ці освітні додатки виявилися корисними для учнів, викладачів та інших зацікавлених сторін.

З метою вдосконалення процесу навчання та забезпечення користувачам зручного і ефективного інструменту для навчання, було розроблено та реалізовано інтерфейс додатку, який сприяє залученню користувача та полегшує засвоєння матеріалу. Під час розробки було використано сучасні методи та технології, такі як макетування, UI/UX дизайн, а також програмування на мовах Python і Kotlin з використанням інструментів розробки Android Studio. Результатом роботи став функціональний та естетичний дизайн додатку, який відповідає вимогам сучасних стандартів розробки програмного забезпечення.

Отриманий додаток має потенціал стати цінним інструментом для студентів та учнів, допомагаючи їм систематизувати та поглибити свої знання. Подальші можливості розвитку включають розширення функціоналу, оптимізацію продуктивності та адаптацію до потреб різних категорій користувачів. Завдяки проведеній роботі вдалося продемонструвати значний потенціал використання інтерактивного дизайну в освітніх додатках, що сприяє підвищенню їхньої ефективності та доступності для широкого кола користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gao, Y.; Chang, C.; Yu, X.; Pang, P.; Xiong, N.; Huang, C. A VR-based volumetric medical image segmentation and visualization system with natural human interaction. *Virtual Real.* 2022, 26, 415.
2. Häfner, P.; Häfner, V.; Ovtcharova, J. Teaching methodology for virtual reality practical course in engineering education. *Procedia Comput. Sci.* 2013, 25, 251.
3. Jia, J.; Chen, W. The Ethical Dilemmas of Virtual Reality Application in Entertainment. In *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, 21–24 July 2017; pp. 699.
4. Johecová, K.; Černý, M.; Stachoň, Z.; Švedová, H.; Káčová, N.; Chmelík, J.; Brůža, V.; Kvarda, O.; Ugwitz, P.; Šašínková, A.; et al. Geography Education in a Collaborative Virtual Environment: A Qualitative Study on Geography Teachers. *ISPRS Int. J. Geo-Inf.* 2022, 11, 180.
5. Wee, C.; Yap, K.M.; Lim, W.N. iProgVR: Design of a Virtual Reality Environment to Improve Introductory Programming Learning; IEEE Access: Piscataway, NJ, USA, 2022; Volume 10, pp.78.
6. Tarng, W.; Chen, C.J.; Lee, C.Y.; Lin, C.M.; Lin, Y.J. Application of virtual reality for learning the material properties of shape memory alloys. *Appl. Sci.* 2019, 9, 580.
7. Al Kork, S.; Beyrouthy, T. Interactive virtual reality educational application. *Adv. Sci. Technol. Eng. Syst. J.* 2018, 3, 282.
8. Freina, L.; Ott, M. A Literature Review on Immersive Virtual Reality in Education: State Of The Art and Perspectives. In *Proceedings of the International Scientific Conference eLearning and Software for Education (eLSE)*, Bucharest, Romania, 23–24 April 2015.
9. Matovu, H.; Ungu, D.; Won, M.; Tsai, C.; Treagust, D.; Mocerino, M.; Tasker, R. Immersive virtual reality for science learning: Design, implementation, and evaluation. *Stud. Sci. Educ.* 2022, 140.

10. Kong, Y. The Role of Experiential Learning on Students' Motivation and Classroom Engagement. *Front. Psychol.* 2021, 12, 272.
11. Yoon, H.J.; Kim, J.; Park, S.W.; Heo, H. Influence of virtual reality on visual parameters: Immersive versus non-immersive mode. *BMC Ophthalmol.* 2020, 20, 200.
12. Chang, E.; Kim, H.-T.; Yoo, B. Virtual Reality Sickness: A Review of Causes and Measurements. *Int. J. Hum.-Comput. Interact.* 2020, 36, 682.
13. Mystakidis, S.; Berki, E.; Valtanen, J.-P. Deep and Meaningful E-Learning with Social Virtual Reality Environments in Higher Education: A Systematic Literature Review. *Appl. Sci.* 2021, 11, 412.
14. Leung, T.; Tkernine, F.; Haruna, I. The use of Virtual Reality in Enhancing Interdisciplinary Research and Education. *arXiv* 2018.
15. Chen, C.J. Theoretical Bases for Using Virtual Reality in Education. *Themes Sci. Technol. Educ.* 2009, 2, 190.
16. Checa, D.; Bustillo, A. A review of immersive virtual reality serious games to enhance learning and training. *Multimed Tools Appl.* 2020, 79, 55.
17. Radianti, J.; Majchrzak, T.A.; Fromm, J.; Wohlgemant, I. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Comput. Educ.* 2019, 147.
18. Kitchenham, B. *Procedures for Performing Systematic Reviews*; Keele University: Keele, UK, 2004; Volume 33.
19. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G.; PRISMA Group. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Med.* 2009, 234.
20. Sedlak, M.; Sasinka, C.; Stachon, Z.; Chmelik, J.; Dolezal, M. Collaborative and individual learning of geography in immersive virtual reality: An effectiveness study. *PLoS ONE* 2022, 17, 224.
21. Meyer, O.; Omdahl, M.K.; Makransky, G. Investigating the effect of pre-training when learning through immersive virtual reality and video: A media and methods experiment. *Comput. Educ.* 2019, 140.

22. Bendeck Soto, J.; Toro Ocampo, D.; Colon, L.; Oropesa, A. Perceptions of ImmerseMe virtual reality platform to improve English communicative skills in higher education. *Int. J. Interact. Mob. Technol. (Ijim)* 2020, 14, 49.
23. Megat, Z.; Megat, A.Z.; Abuhassna, H. Virtual reality acceptance in classrooms: A case study in teaching science. *Int. J. Adv. Trends Comput. Sci. Eng.* 2020, 9, 125.
24. Xu, X.; Ke, F. Designing a virtual-reality-based, gamelike math learning environment. *Am. J. Distance Educ.* 2016, 30, 738.
25. Zhang, K.; Suo, J.; Chen, J.; Liu, X.; Gao, L. Design and implementation of fire safety education system on campus based on virtual reality technology. In *Proceedings of the Position Papers of the 2017 Federated Conference on Computer Science and Information Systems, Prague, Czech Republic, 3–6 September 2017; Volume 11, pp. 129.*
26. Kamińska, D.; Sapiński, T.; Aitken, N.; Rocca, A.; Barańska, M.; Wietsma, R. Virtual reality as a new trend in mechanical and electrical engineering education. *J. Phys.* 2017, 15, 936.
27. Li, Y.; Ying, S.; Chen, Q.; Guan, J. An Experiential Learning-Based Virtual Reality Approach to Foster Students' Vocabulary Acquisition and Learning Engagement in English for Geography. *Sustainability* 2022, 14, 153.
28. Pande, P.; Thit, A.; Sørensen, A.E.; Mojsoska, B.; Moeller, M.E.; Jepsen, P.M. Long-term effectiveness of immersive VR simulations in undergraduate science learning: Lessons from a media-comparison study. *Res. Learn. Technol.* 2021, 29.
29. Pirker, J.; Lesjak, I.; Guetl, C. An educational physics laboratory in mobile versus room scale virtual reality—A comparative study. *Int. J. Online Eng. (Ijoe)* 2017, 13, 106.
30. Akman, E.; Çakır, R. Pupils' opinions on an educational virtual reality game in terms of flow experience. *Int. J. Emerg. Technol. Learn. (Ijet)* 2019, 14, 121.
31. Chang, S.C.; Hsu, T.; Chen, Y.N.; Morris, J. The effects of spherical video-based virtual reality implementation on students' natural science learning effectiveness. *Interact. Learn. Environ.* 2018

ДОДАТКИ

Додаток А. Лістинг програми

```
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_sqlalchemy import SQLAlchemy

from flask_login import LoginManager, login_user, login_required,
logout_user, current_user

from werkzeug.security import generate_password_hash, check_password_hash
from models import db, User, Course, Enrollment

app = Flask(__name__)
app.config['SECRET_KEY'] = 'your_secret_key'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///site.db'
db.init_app(app)

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))

# Створюємо таблиці, якщо їх ще не існує
with app.app_context():
    db.create_all()
```

```

@app.route('/')
@login_required
def index():
    courses = Course.query.filter_by(user_id=current_user.id).all()
    return render_template('index.html', courses=courses)

```

```

@app.route('/add_course', methods=['GET', 'POST'])
@login_required
def add_course():
    if request.method == 'POST':
        course_name = request.form['course_name']
        if course_name:
            new_course = Course(name=course_name, user_id=current_user.id)
            db.session.add(new_course)
            db.session.commit()
            return redirect(url_for('index'))
    return render_template('add_course.html')

```

```

@app.route('/edit_course/<int:course_id>', methods=['GET', 'POST'])
@login_required
def edit_course(course_id):
    course = Course.query.get_or_404(course_id)
    if course.user_id != current_user.id:
        return redirect(url_for('index'))

```

```

if request.method == 'POST':
    course.name = request.form['course_name']
    db.session.commit()
    return redirect(url_for('index'))

return render_template('edit_course.html', course=course)

```

```
@app.route('/delete_course/<int:course_id>', methods=['POST'])
```

```
@login_required
```

```
def delete_course(course_id):
```

```
    course = Course.query.get_or_404(course_id)
```

```
    if course.user_id != current_user.id:
```

```
        return redirect(url_for('index'))
```

Видаляємо записи про записи на курс, які посилаються на видаляємий курс

```
enrollments = Enrollment.query.filter_by(course_id=course.id).all()
```

```
for enrollment in enrollments:
```

```
    db.session.delete(enrollment)
```

```
db.session.commit()
```

```
# Тепер видаляємо сам курс
```

```
db.session.delete(course)
```

```
db.session.commit()
```

```
return redirect(url_for('index'))
```

```
@app.route('/register', methods=['GET', 'POST'])
```

```
def register():
```

```

if request.method == 'POST':
    username = request.form['username']
    password = request.form['password']
    user = User.query.filter_by(username=username).first()
    if user:
        flash('Ім'я користувача вже існує. Будь ласка, виберіть інше.')
        return redirect(url_for('register'))
    hashed_password = generate_password_hash(password,
method='pbkdf2:sha256')
    new_user = User(username=username, password=hashed_password)
    db.session.add(new_user)
    db.session.commit()
    flash('Реєстрація пройшла успішно!')
    return redirect(url_for('login'))
return render_template('register.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        user = User.query.filter_by(username=username).first()
        if user and check_password_hash(user.password, password):
            login_user(user)
            return redirect(url_for('index'))
        flash('Невірне ім'я користувача або пароль')
    return render_template('login.html')

@app.route('/logout')
@login_required

```

```

def logout():
    logout_user()
    return redirect(url_for('login'))

@app.route('/enroll/<int:course_id>', methods=['POST'])
@login_required
def enroll(course_id):
    course = Course.query.get_or_404(course_id)
    enrollment = Enrollment(user_id=current_user.id, course_id=course.id)
    db.session.add(enrollment)
    db.session.commit()
    return redirect(url_for('index'))

@app.route('/user_info')
@login_required
def user_info():
    user = User.query.get_or_404(current_user.id)
    enrollments = Enrollment.query.filter_by(user_id=user.id).all()
    return render_template('user_info.html', user=user, enrollments=enrollments)

if __name__ == '__main__':
    app.run(debug=True)

```